

12



Industria y Comercio

SUPERINTENDENCIA

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO



No. 14-060246-00000-0000

Fecha: 2014-03-20 13:45:34 Dep. 2020 DIR.NUEVASCR
Tra. 3 MODELO Eve: 1 REGDEPOSITO
Act. 411 PRESENTACION Folios: 90

DELEGATURA DE PROPIEDAD INDUSTRIAL

División de Nuevas Creaciones

SOLICITUD

PATENTE DE MODELO DE UTILIDAD

21. EXPEDIENTE No. _____

54. TÍTULO MICRO PUBLIK

51. CLASIFICACIÓN INTERNACIONAL _____

71. SOLICITANTE PUBLIK TECNOLOGIAS INFORMACION
COMUNICACION Y MEDIO AMBIENTE SAS
DOMICILIO AV 30 AGOSTO 87,-362, PEREIRA.

74. APODERADO _____

22. BOGOTÁ, D. C., 20/ MARZO /2014.

Industria y Comercio
SUPERINTENDENCIA

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

o de radicación



No. 14-060246- -00000-0000

Fecha: 2014-03-20 13:45:34 Dep. 2020 DIR.NUEVASCR
Tra. 3 MODELO Eve: 1 REGDEPOSITO
Act. 411 PRESENTACION Folios: 90**FORMULARIO ÚNICO DE SOLICITUD DE PATENTE****SOLICITUD DE:**

①

☐ Patente de Invención☒ Patente de Modelo de Utilidad

②

SOLICITANTENombre: Publik Tecnologias Informacion
Comunicaciones y Medio Ambiente S.A.S
Dirección: Av. 30 de Agosto N° 87-362
Nacionalidad o Domicilio: Colombiana
Lugar de Constitución: Pereira – Risaralda-Colombia
Teléfono: (6)3377500 Fax:
E- mail: publik@publik.co**IDENTIFICACIÓN**☐ C.C.☒ NIT☐ C.E.☐ Otro

Cúal

Número 891.902.435-4

③

**REPRESENTANTE O
APODERADO**Nombre: Jorge Enrique Giraldo Nieto.
Dirección: Av. 30 de Agosto N° 87-362 Pereira
Teléfono: (6) 3377500 Fax:
E- mail: publik@publik.co**IDENTIFICACIÓN**☒ C.C.☐ NIT☐ C.E.☐ Otro

Cúal

Número 7'517.585

TP

En caso de haber presentado poder general para asuntos que se adelanten ante la Delegatura de
Propiedad Industrial, sírvase indicar el numero de radicación (del poder).

④

INVENTOR (ES)
(72)Nombre: Luis Alejandro Hernández Carrillo
Dirección: Calle 17 B N° 9N-28 Prado Alto Cartago (Valle)
Nacionalidad o Domicilio: Colombiano

⑤

Título (54)**MICRO PUBLIK** es una tarjeta controladora que permite transmitir mensajes a

informadores o equipos de luz incandescente.

⑥

Número de reivindicaciones

1.

⑦

Clasificación Internacional (51)

⑧

Prioridad☐ SI☐ NO

(33)País de Origen

(32) Fecha

(31) Número de Solicitud

⑨

Para publicar a partir de la fecha de la presente solicitud o de la prioridad invocada a los:

☐ Inmediato☐ 6 meses☐ 12 meses☐ 18 meses

⑩

Comprobante de pago No. 64975319-3

Fecha

18/03/14

11

Anexos

- ☐ Comprobante de pago de la tasa de presentación de la solicitud.
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☐ Comprobante de pago por reivindicación de prioridad.
- ☐ Documento que acredite la existencia y representación legal de la persona jurídica peticionaria, en cuanto no estén inscritas en alguna Cámara de Comercio Colombiana.
- ☐ Poder, si fuere el caso.
- ☐ Certificado de la fecha de presentación de la solicitud prioridad expedida por la autoridad correspondiente y una copia certificada de la primera solicitud, si se reivindica prioridad
- ☐ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☐ Resumen
- ☐ Descripción de la invención.
- ☐ Una o más reivindicaciones.
- ☐ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico
- ☐ Arte final 12x12 cm.
- ☐ De ser el caso, información sobre otras solicitudes de patente o títulos obtenidos en el extranjero por el mismo titular o su causante, relacionadas parcial o totalmente con la invención de esta solicitud.

12

FIGURA CARACTERÍSTICA:

13

Solicito la concesión de la patente,

NOMBRE: JORGE ENRIQUE GIRALDO NIETO

FIRMA:

C.C. 7'517.585 TP

Instrucciones para la presentación de los anexos, ver reverso de esta página.

43

CAMARA DE COMERCIO DE PEREIRA

No 4480567

CERTIFICADO DE EXISTENCIA Y REPRESENTACION LEGAL

Número de operación: 01AF70312011 Fecha: 20140312 Hora: 10:42:39 Pagina: 1

CERTIFICADO DE EXISTENCIA Y REPRESENTACION LEGAL O INSCRIPCION DE DOCUMENTOS.

LA CAMARA DE COMERCIO DE PEREIRA , CON FUNDAMENTO EN LAS MATRICULAS E INSCRIPCIONES DEL REGISTRO MERCANTIL,

CERTIFICA :

NOMBRE : PUBLIK TECNOLOGIAS INFORMACION COMUNICACIONES Y MEDIO AMBIENTE S.A.S

N.I.T.: 891902435-4

DIRECCION COMERCIAL: AVENIDA 30 DE AGOSTO NRO. 87 362

DOMICILIO : PEREIRA

TELEFONO COMERCIAL 1: 3377500

DIRECCION DE NOTIFICACION JUDICIAL : AVENIDA 30 DE AGOSTO NRO. 87 362

DIRECCION PAGINA WEB (URL) : WWW.PUBLIK.COM.CO

MUNICIPIO JUDICIAL: PEREIRA

E-MAIL COMERCIAL: publik@publik.com.co

E-MAIL NOT. JUDICIAL: publik@publik.com.co

TELEFONO NOTIFICACION JUDICIAL 1: 3377500

FAX NOTIFICACION JUDICIAL:

ACTIVIDADES ECONOMICAS:

ACTIVIDAD PRINCIPAL:

7310 PUBLICIDAD

ACTIVIDAD SECUNDARIA:

2610 FABRICACION DE COMPONENTES Y TABLEROS ELECTRONICOS

ACTIVIDAD ADICIONAL 1:

7730 ALQUILER Y ARRENDAMIENTO DE OTROS TIPOS DE MAQUINARIA, EQUIPO Y BIENES TANGIBLES N.C.P.

CERTIFICA :

MATRICULA NO. 06149812 'A F I L I A D O'

FECHA DE MATRICULA EN ESTA CAMARA: 14 DE FEBRERO DE 1991

RENOVO EL AÑO 2013 , EL 22 DE MARZO DE 2013

CERTIFICA :

CONSTITUCION : QUE POR ESCRITURA PUBLICA NO. 0001191 DE NOTARIA SEGUNDA DE PEREIRA DEL 31 DE JULIO DE 1979 , INSCRITA EL 14 DE FEBRERO DE 1991 BAJO EL NUMERO 00910154 DEL LIBRO IX, SE CONSTITUYO LA PERSONA JURIDICA: VALTRONIK COMPAÑIA LIMITADA QUE POR ESCRITURA PUBLICA NO. 0003202 DE NOTARIA PRIMERA DE PEREIRA DEL 30 DE JULIO DE 1997 , INSCRITA EL 4 DE AGOSTO DE 1997 BAJO EL NUMERO 00005710 DEL LIBRO IX, LA SOCIEDAD CAMBIO SU NOMBRE DE : VALTRONIK COMPAÑIA LIMITADA POR EL DE : TECNOLOGIA Y COMUNICACION VALTRONIK S.A.

QUE POR ACTA NO. 0000080 DE ASAMBLEA DE ACCIONISTAS DE PEREIRA DEL 29 DE ENERO DE 2010 , INSCRITA EL 20 DE DICIEMBRE DE 2010 BAJO EL NUMERO 01017873 DEL LIBRO IX, LA SOCIEDAD CAMBIO SU NOMBRE DE : TECNOLOGIA Y COMUNICACION VALTRONIK S.A. POR EL DE : PUBLIK TECNOLOGIAS INFORMACION COMUNICACIONES Y MEDIO AMBIENTE S.A.S

CERTIFICA :

QUE POR ESCRITURA PUBLICA NO. 0003202 DE NOTARIA PRIMERA DE PEREIRA DEL 30 DE JULIO DE 1997 , INSCRITA EL 4 DE AGOSTO DE 1997 BAJO EL NUMERO 00005710 DEL LIBRO IX, LA PERSONA JURIDICA TUVO LA SIGUIENTE TRANSFORMACION : DE SOCIEDAD LTDA. A SOCIEDAD ANONIMA

QUE POR ACTA NO. 0000080 DE ASAMBLEA DE ACCIONISTAS DE PEREIRA DEL 29 DE ENERO DE 2010 , INSCRITA EL 20 DE DICIEMBRE DE 2010 BAJO EL NUMERO 01017873 DEL LIBRO IX, LA PERSONA JURIDICA TUVO LA SIGUIENTE TRANSFORMACION : DE SOCIEDAD ANONIMA A SOCIEDAD POR ACCIONES SIMPLIFICADA, BAJO LA RAZON SOCIAL DE: PUBLIK TECNOLOGIAS INFORMACION COMUNICACIONES Y MEDIO AMBIENTE S.A.S.LA SOCIEDAD SE IDENTIFICARA CON EL NOMBRE COMERCIAL: PUBLIK

CERTIFICA :

REFORMAS:

DOCUMENTO	FECHA	ORIGEN	CIUDAD	INSCRIPCION	FECHA
0001898	1981/09/18	NOTARIA SEGUNDA	PER	00910155	1991/02/14
0002967	1983/12/19	NOTARIA SEGUNDA	PER	00910156	1991/02/14
0003131	1984/12/20	NOTARIA SEGUNDA	PER	00910157	1991/02/14
0006026	1986/12/29	NOTARIA PRIMERA	PER	00910158	1991/02/14
0000492	1991/01/28	NOTARIA PRIMERA	PER	00910159	1991/02/14
0003157	1992/06/25	NOTARIA PRIMERA	PER	00920916	1992/07/02
0003233	1995/07/28	NOTARIA PRIMERA	PER	00002618	1995/08/04
0005775	1996/12/06	NOTARIA PRIMERA	PER	00004811	1996/12/06
0003202	1997/07/30	NOTARIA PRIMERA	PER	00005710	1997/08/04
0004954	1998/11/12	NOTARIA PRIMERA	PER	00007475	1998/11/18
0002292	2002/06/07	NOTARIA PRIMERA	PER	00012379	2002/06/14
0000000	2002/06/28	REVISOR FISCAL	PER	00012470	2002/07/04
0000000	2002/10/07	REVISOR FISCAL	PER	00012909	2002/10/11
0003955	2007/07/27	ASAMBLEA DE ACCIONISPER		01009590	2007/08/01
0000206	2008/01/17	ASAMBLEA GENERAL DE ACC		01010658	2008/01/22
0000080	2010/01/29	ASAMBLEA DE ACCIONISPER		01017873	2010/12/20
0000086	2011/05/11	ASAMBLEA GENERAL EXTRAO		01019016	2011/05/13

CERTIFICA

VIGENCIA: QUE EL TERMINO DE DURACION DE LA PERSONA JURIDICA ES INDEFINIDO

CERTIFICA :

OBJETO SOCIAL.- LA SOCIEDAD REALIZARA LAS SIGUIENTES ACTIVIDADES: 1. DISEÑO, PRODUCCION, OPERACION, PROGRAMACION, DIFUSION EN TODO TIPO DE SISTEMAS DE INFORMACION COMO MEDIO MASIVO DE COMUNICACION. 2. DISEÑO, INTERVENTORIA, CONSULTORIA, EJECUCION, PRODUCCION, DISTRIBUCION, COMERCIALIZACION, IMPORTACION Y EXPORTACION DE TODO TIPO DE PROYECTOS, OBRAS,

SERVICIOS RELACIONADOS CON LA ELECTRICA, ELECTRONICA, TELEMATICA, INFORMATICA, MEDIO AMBIENTE Y TELECOMUNICACIONES, INCLUIDO EL DESARROLLO DE SOFTWARE, HARDWARE, DATAWARE Y PLATAFORMAS INTERACTIVAS. 3. LA PRESTACION DE SERVICIOS DE CONSULTORIA E INTERVENTORIA EN TODO TIPO DE OBRAS Y PROYECTOS RELACIONADOS CON LOS NUMERALES ANTERIORES. 4. CUALQUIER ACTIVIDAD LICITA CIVIL O COMERCIAL.

CERTIFICA :

CAPITAL:

** CAPITAL AUTORIZADO **

VALOR :\$1,000,000,000.00

NO. DE ACCIONES:10,000.00

VALOR NOMINAL :\$100,000.00

** CAPITAL SUSCRITO **

VALOR :\$600,000,000.00

NO. DE ACCIONES:6,000.00

VALOR NOMINAL :\$100,000.00

** CAPITAL PAGADO **

VALOR :\$600,000,000.00

NO. DE ACCIONES:6,000.00

VALOR NOMINAL :\$100,000.00

CERTIFICA :

** JUNTA DIRECTIVA: PRINCIPAL(ES) **

QUE POR ESCRITURA PUBLICA NO. 0003202 DE NOTARIA PRIMERA DE PEREIRA DEL 30 DE JULIO DE 1997 , INSCRITA EL 4 DE AGOSTO DE 1997 BAJO EL NUMERO 00005710 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE

IDENTIFICACION

MIEMBRO PRINCIPAL JUNTA DIRECTIVA

GIRALDO NIETO JORGE ENRIQUE

C.C.00007517585

MIEMBRO PRINCIPAL JUNTA DIRECTIVA

SARABIA DE GIRALDO CARMEN ASTRID

C.C.00024954543

QUE POR ACTA NO. 0000082 DE ASAMBLEA EXTRAORDINARIA DE ACCIO DEL 11 DE OCTUBRE DE 2010 , INSCRITA EL 21 DE OCTUBRE DE 2010 BAJO EL NUMERO 01017420 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE

IDENTIFICACION

MIEMBRO PRINCIPAL JUNTA DIRECTIVA

GIRALDO SARABIA LAURA MARIA

C.C.00044001499

** JUNTA DIRECTIVA: SUPLENTE(S) **

QUE POR ACTA NO. 0000090 DE ACCIONISTAS CONSTITUYENTES DEL 26 DE MAYO DE 2012 , INSCRITA EL 27 DE JUNIO DE 2012 BAJO EL NUMERO 01027216 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE

IDENTIFICACION

MIEMBRO SUPLENTE JUNTA DIRECTIVA

MONTTOYA GALEANO JACKELINE

C.C.00042109670

QUE POR ACTA NO. 0000056 DE ASAMBLEA DE ACCIONISTAS DEL 23 DE OCTUBRE DE 2001 , INSCRITA EL 14 DE NOVIEMBRE DE 2001 BAJO EL NUMERO 00011510 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE

IDENTIFICACION

MIEMBRO SUPLENTE JUNTA DIRECTIVA

GIRALDO SARABIA JUAN PABLO

C.C.00071386515

QUE POR ACTA NO. 0000082 DE ASAMBLEA EXTRAORDINARIA DE ACCIO DEL 11 DE OCTUBRE DE 2010 , INSCRITA EL 21 DE OCTUBRE DE 2010 BAJO EL NUMERO 01017420 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE
MIEMBRO SUPLENTE JUNTA DIRECTIVA
OSORIO BRAND LORENA

IDENTIFICACION
C.C.00032296218

CERTIFICA :

**** NOMBRAMIENTOS : ****

QUE POR ESCRITURA PUBLICA NO. 0003202 DE NOTARIA PRIMERA DE PEREIRA DEL 30 DE JULIO DE 1997 , INSCRITA EL 4 DE AGOSTO DE 1997 BAJO EL NUMERO 00005710 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE IDENTIFICACION
PRESIDENTE

GIRALDO NIETO JORGE ENRIQUE C.C.00007517585
QUE POR ESCRITURA PUBLICA NO. 0003955 DE NOTARIA PRIMERA DE PEREIRA DEL 27 DE JULIO DE 2007 , INSCRITA EL 1 DE AGOSTO DE 2007 BAJO EL NUMERO 01009591 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE IDENTIFICACION
VICEPRESIDENTE

SARABIA DE GIRALDO CARMEN ASTRID C.C.00024954543
QUE POR ACTA NO. 0000087 DE ASAMBLEA GENERAL EXTRAORDINARIA DEL 1 DE AGOSTO DE 2011 , INSCRITA EL 11 DE NOVIEMBRE DE 2011 BAJO EL NUMERO 01020353 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE IDENTIFICACION
GERENTE

MONTOYA GALEANO JACKELINE C.C.00042109670
QUE POR ACTA NO. 0000007 DE JUNTA DIRECTIVA DEL 26 DE MAYO DE 2012 , INSCRITA EL 27 DE JUNIO DE 2012 BAJO EL NUMERO 01027217 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE IDENTIFICACION
SUBGERENTE

GIRALDO SARABIA LAURA MARIA C.C.00044001499

CERTIFICA :

REPRESENTACION LEGAL.- LA REPRESENTACION LEGAL DE LA SOCIEDAD ESTARA A CARGO DE UNA PERSONA NATURAL DENOMINADA PRESIDENTE, O DE UNA JURIDICA, ACCIONISTAS O NO, CON SUS RESPECTIVOS SUPLENTE, DESIGNADOS PARA UN TERMINO DE UN AÑO POR LA JUNTA DIRECTIVA. LAS FUNCIONES DEL REPRESENTANTE LEGAL TERMINARAN EN CASO DE DIMISION O REVOCACION POR PARTE DE LA JUNTA DIRECTIVA, DE DECESO O DE INCAPACIDAD EN AQUELLOS CASOS EN QUE EL REPRESENTANTE LEGAL SEA UNA PERSONA NATURAL, Y EN CASO DE LIQUIDACION PRIVADA O JUDICIAL, CUANDO EL REPRESENTANTE LEGAL SEA UNA PERSONA JURIDICA. LA REVOCACION O REMOCION POR PARTE DE LA JUNTA DIRECTIVA NO TENDRA QUE ESTAR MOTIVADA Y PODRA REALIZARSE EN CUALQUIER TIEMPO. FACULTADES Y RESTRICCIONES: LA SOCIEDAD SERA GERENCIADA, ADMINISTRADA Y REPRESENTADA LEGALMENTE ANTE TERCEROS POR UNO O VARIOS REPRESENTANTES LEGALES. SERAN DESIGNADOS POR LA JUNTA DIRECTIVA O POR LA ASAMBLEA GENERAL DE ACCIONISTAS POR PERIODOS ANUALES QUE SE PODRAN RENOVAR INDEFINIDAMENTE. ESTOS REPRESENTANTES LEGALES PODRAN REALIZAR CUALQUIER CLASE DE NEGOCIACION Y CELEBRAR CUALQUIER CLASE DE CONTRATO. LOS REPRESENTANTES LEGALES TENDRAN LAS SIGUIENTES FACULTADOS: 1. ASISTIR A LAS REUNIONES DE JUNTA DIRECTIVA CON VOZ Y SIN VOTO, A MENOS QUE SEAN MIEMBROS DE LA JUNTA DIRECTIVA. 2. REPRESENTAR A LA SOCIEDAD ANTE LOS ACCIONISTAS Y ANTE TERCEROS. 3. EJECUTAR TODOS LOS ACTOS ENCAMINADOS A CUMPLIR CON

LAS DIRECTRICES DE LA ASAMBLEA GENERAL DE ACCIONISTAS Y LA JUNTA DIRECTIVA. 4. CONVOCAR A LA ASAMBLEA GENERAL DE ACCIONISTAS PARA REUNIONES DE CARACTER ORDINARIO O EXTRAORDINARIO. 5. CONVOCAR A LA JUNTA DIRECTIVA CUANDO LA ESTIMO CONVENIENTE. RESTRCCIONES: A. EL PRESIDENTE Y EL VICEPRESIDENTE NO TENDRAN NINGUNA CLASE DE RESTRICCIÓN EN LA REPRESENTACIÓN LEGAL DE LA SOCIEDAD. B. EL GERENTE SOLO PODRA SUSCRIBIR ACTOS O CONTRATOS EN NOMBRE DE LA SOCIEDAD HASTA POR UNA CUANTIA MAXIMA DE CINCUENTA (50 SMLMV) SALARIOS MINIMOS LEGALOS MENSUALS VIGENTES. C. EL SUB-GERENTE SOLO PODRA EJERCER LA REPRESENTACIÓN LEGAL EN LOS SIGUIENTES CASOS: 1. PARA ACTUAR EN LOS PROCESOS LICITATORIOS CONVOCADOS POR LAS ENTIDADES PUBLICAS, MIXTAS O PRIVADAS NACIONALES E INTERNACIONALES. 2. PARA QUE SE NOTIFIQUE DE CUALQUIER PROVIDENCIA JUDICIAL O ADMINISTRATIVA, EN LA CUAL DEBA COMPARECER NUESTRA SOCIEDAD BIEN SEA COMO. DEMANDANTE, DEMANDADA, ACCIONANTE, O ACCIONADA. D) SE PROHIBE TOTALMENTE AL GERENTE Y AL SUBGERENTE LA ENAJENACIÓN DE ACTIVOS FIJOS PRODUCTIVOS Y EL ENDEUDAMIENTO DE LA SOCIEDAD. ESTAS RESTRICCIONES PODRAN SER LEVANTADAS TEMPORALMENTE POR LA JUNTA DIRECTIVA.

CERTIFICA :

** REVISOR FISCAL: **

QUE POR ESCRITURA PUBLICA NO. 0003202 DE NOTARIA PRIMERA DE PEREIRA DEL 30 DE JULIO DE 1997 , INSCRITA EL 4 DE AGOSTO DE 1997 BAJO EL NUMERO 00005710 DEL LIBRO IX , FUE(ON) NOMBRADO(S):
NOMBRE IDENTIFICACION

REVISOR FISCAL PRINCIPAL -T.P. 4453-T

BUILES YEPES HERNANDO DE JESUS

C.C.00003351886

QUE POR ACTA NO. 0000072 DE ASAMBLEA DE ACCIONISTAS DEL 8 DE MARZO DE 2007 , INSCRITA EL 25 DE ABRIL DE 2007 BAJO EL NUMERO 01008979 DEL LIBRO IX , FUE(ON) NOMBRADO(S):

NOMBRE

IDENTIFICACION

REVISORA FISCAL SUPLENTE -T.P. 21658-T

GONZALEZ TABARES LUZ STELLA

C.C.00031196258

CERTIFICA :

QUE LA PERSONA JURIDICA TIENE MATRICULADOS LOS SIGUIENTES ESTABLECIMIENTOS :

NOMBRE : PUBLIK

MATRICULA NO. 06149902 DEL 14 DE FEBRERO DE 1991

RENOVACION DE LA MATRICULA : EL 22 DE MARZO DE 2013

ULTIMO AÑO RENOVADO : 2013

ACTIVIDADES ECONOMICAS:

ACTIVIDAD PRINCIPAL:

7310 PUBLICIDAD

ACTIVIDAD SECUNDARIA:

2610 FABRICACION DE COMPONENTES Y TABLEROS ELECTRONICOS

ACTIVIDAD ADICIONAL 1:

7730 ALQUILER Y ARRENDAMIENTO DE OTROS TIPOS DE MAQUINARIA,

EQUIPO Y BIENES TANGIBLES N.C.P.

CERTIFICA :

QUE NO FIGURAN INSCRIPCIONES ANTERIORES A LA FECHA DEL PRESENTE CERTIFICADO, QUE MODIFIQUEN TOTAL O PARCIALMENTE SU CONTENIDO.

CERTIFICA :

06353303

DE ACUERDO CON LO PREVISTO EN LA LEY 962 DE 2005, LOS ACTOS DE INSCRIPCION QUE SE CERTIFIQUEN, QUEDAN EN FIRME DIEZ (10) DIAS DESPUES DE SU REGISTRO, SIEMPRE Y CUANDO NO SEAN OBJETO DE LOS RECURSOS EN LA VIA GUBERNATIVA (REPOSICION, APELACION O QUEJA)

***** I N F O R M A *****

LA CÁMARA DE COMERCIO DE PEREIRA HA EFECTUADO MIGRACIÓN DE LA INFORMACIÓN DE LOS REGISTROS PÚBLICOS A UN NUEVO SISTEMA REGISTRAL, LO CUAL PUEDE OCASIONAR OMISIONES O ERRORES EN LA INFORMACIÓN CERTIFICADA, NO OBSTANTE EL CONTROL DE CALIDAD EFECTUADO POR LA ENTIDAD; POR LO CUAL EN CASO DE ENCONTRAR ALGUNA OBSERVACIÓN EN ESTE CERTIFICADO, VERIFICAREMOS LA INFORMACIÓN Y PROCEDEREMOS A SU REEMPLAZO.

CERTIFICADO SIN VALOR PARA EL AFILIADO

DE CONFORMIDAD CON EL DECRETO 2150 DE 1995 Y LA AUTORIZACION IMPARTIDA POR LA SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO, MEDIANTE EL OFICIO DEL 18 DE NOVIEMBRE DE 1996, LA FIRMA MECANICA QUE APARECE A CONTINUACION TIENE PLENA VALIDEZ PARA TODOS LOS EFECTOS LEGALES

Mamiolegal

CESIÓN DE DERECHOS PATRIMONIALES

76

Entre nosotros: **JACKELINE MONTOYA GALEANO**, Mayor de edad, con domicilio en la ciudad de Pereira, identificada con cedula de ciudadanía N°. 42.109.670 expedida en Pereira, obrando en calidad de representante legal de **PUBLIK TECNOLOGIAS INFORMACION COMUNICACIONES Y MEDIO AMBIENTE S.A.S.**, con domicilio en la ciudad de Pereira NIT 891.902.435-4, quien para los efectos de este contrato se denominará EL CESIONARIO, por una parte y por la otra, **LUIS ALEJANDRO HERNANDEZ CARRILLO**, mayor de edad, identificado con la cedula No 1.110.460.048 expedida en Ibagué Tolima, quien en adelante se denominará **EL AUTOR O CEDENTE**, se ha acordado suscribir el presente contrato de Cesión de Derechos de Autor, el cual se regirá por la ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia y en especial por las siguientes cláusulas:

PRIMERA- OBJETO: EL AUTOR manifiesta que de manera voluntaria y gratuita, realiza la Cesión a favor de EL CESIONARIO, de todos los derechos de autor patrimoniales que a él le corresponden como creador de la **HARDWARE** denominado **TARJETA MICROPUBLIK** cuyas características son las siguientes:

La tarjeta MicroPublik es una tarjeta Inteligente que permite transmitir mensajes en un display electrónico, cuya etapa de potencia esté constituida por las tarjetas P790 o las Board Driver de los informadores Isigma. Esta tarjeta permite visualizar mensajes con cierta variedad de efectos que son: Rotate, Hold, Flash, Scroll, WipeUp. Estos efectos pueden ser programados con 10 velocidades distintas de visualización.

Esta tarjeta cuenta con otras funciones, como visualización de temperatura, hora en 2 formatos, fecha en tres formatos y texto en negrilla. La programación de los mensajes y las funciones mencionadas, se realizan vía puerto serial RS485 ó RS232 (red celular) desde un software de programación, llamado Datasoft Mensajes Info, el cual también se utiliza para programar las tarjetas Microcinta.

Esta tarjeta se realizó de tal manera que tuviera un diseño compacto y con cierta autonomía, la cual se logra mediante su propia fuente de alimentación de 12 y 5 voltios, incorporada.

SEGUNDA- ALCANCE DEL OBJETO: Los derechos aquí cedidos comprenden todos los derechos patrimoniales (Reproducción, transformación, comunicación pública, distribución y cualquier otro que se derive de la utilización de la obra que represente un beneficio económico) y se dan sin limitación alguna en cuanto a territorio se refiere; esta Cesión se da por todo el término de duración establecido en la legislación autoral vigente en Colombia.

CESIÓN DE DERECHOS PATRIMONIALES

87



TERCERA- DERECHOS MORALES: EL CESIONARIO se compromete con el AUTOR/CEDENTE a respetar los derechos morales sobre la obra que le corresponden como autor en los términos del artículo 30 de la Ley 23 de 1982 y Artículo 11 de la Decisión Andina 351 de 1993.

CUARTA- EL AUTOR O CEDENTE, manifiesta que la obra objeto del presente contrato es original y fue realizada por él sin violar o usurpar derechos de autor de terceros, por lo tanto la obra es de su exclusiva autoría y detenta la titularidad de la misma, la cual cede en virtud del presente contrato.

QUINTA- EXCLUSIVIDAD: EL AUTOR O CEDENTE manifiesta que los derechos sobre la obra en cuestión no han sido cedidos con antelación y que sobre ellos no pesa ningún gravamen ni limitación en su uso o utilización.

Para constancia se firma el presente documento a los (17) diecisiete días del mes de marzo de 2014

EL AUTOR

EL CESIONARIO

LUIS ALEJANDRO HERNANDEZ CARRILLO
T.P. QN206-76603
Ingeniero Electrónico

JACKELINE MONTOYA GALEANO
Representante Legal
PUBLIK S.A.S.

**NOTARIA PRIMERA DEL CIRCULO DE PEREIRA
DILIGENCIA DE RECONOCIMIENTO**

Ante mí, JOSE DANIEL TRUJILLO ARCILA, Notario Primero,
Compareció: Yacheline Montoya Galeano

Con C.C. 42109636

y declaró que la firma y huella que aparece en el presente documento es suya y que el contenido del mismo es cierto



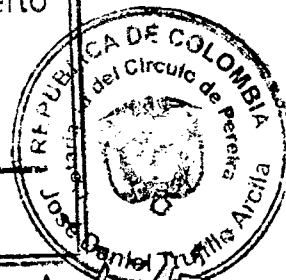
Huella Dactilar

El Compareciente

Yacheline Montoya Galeano

Fecha.

18 MAR 2014



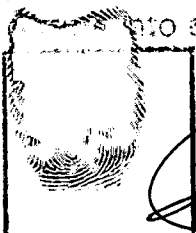
✓ Yacheline Montoya Galeano

**NOTARIA PRIMERA DEL CIRCULO DE PEREIRA
DILIGENCIA DE RECONOCIMIENTO**

Ante mí, JOSE DANIEL TRUJILLO ARCILA, Notario Primero,
Compareció: LUIS Alejandro Hernandez

Con C.C. 94411071110460048

y declaró que la firma y huella que aparece en el presente documento es suya y que el contenido del mismo es cierto



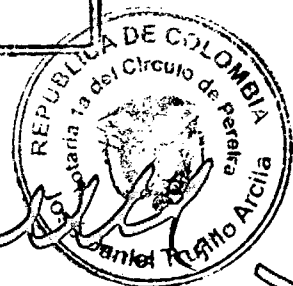
Huella Dactilar

El Compareciente

Luis Alejandro Hernandez

Fecha.

18 MAR 2014



✓ Luis Alejandro Hernandez

90

RESUMEN

La tarjeta MicroPublik es una tarjeta capaz de transmitir mensajes en un display electrónico, cuya etapa de potencia esté constituida por las tarjetas o las Board Driver de los informadores o equipos de luz incandescente. Esta tarjeta permite visualizar mensajes con cierta variedad de efectos que son: Rotate, Hold, Flash, Scroll, WipeUp. Estos efectos pueden ser programados con 10 velocidades distintas de visualización.

Esta tarjeta cuenta con otras funciones, como visualización de temperatura, hora en 2 formatos, fecha en tres formatos y texto en negrilla.

Esta tarjeta se realizó de tal manera que tuviera un diseño compacto y con cierta autonomía, la cual se logra mediante su propia fuente de alimentación de 12 y 5 voltios, incorporada.

DESCRIPCIÓN TARJETA MICROPUBLIK

INTRODUCCIÓN

La tarjeta Micropublik, es una tarjeta controladora que permite transmitir mensajes a informadores o equipos incandescentes. Esta tarjeta permite visualizar mensajes con 5 tipos de efectos que son:

- Rotate
- Hold
- Flash
- Scroll
- WipeUp

Otras características son.

- Posibilidad de variar la velocidad en cada efecto
- Letra en Negrilla
- Hora Militar/AMPM
- 3 formatos de Fecha
- Temperatura
- Control de intensidad
- Posibilidad de manejar más de 500 verticales.

DESCRIPCIÓN

La tarjeta MicroPublik es una tarjeta capaz de transmitir mensajes en un display electrónico, cuya etapa de potencia esté constituida por las tarjetas o las Board Driver de los informadores o equipos incandescentes. Esta tarjeta permite visualizar mensajes con cierta variedad de efectos que son: Rotate, Hold, Flash, Scroll, WipeUp. Estos efectos pueden ser programados con 10 velocidades distintas de visualización.

Esta tarjeta cuenta con otras funciones, como visualización de temperatura, hora en 2 formatos, fecha en tres formatos y texto en negrilla. La programación de los mensajes y las funciones mencionadas, se realizan vía puerto serial RS485 ó RS232 desde un software de programación.

La tarjeta se realizó de tal manera que tuviera un diseño compacto y con cierta autonomía, la cual se logra mediante su propia fuente de alimentación de 12 y 5 voltios, incorporada.

Los componentes principales de la tarjeta son:

- Memoria SRAM no volátil, de 32 Kbytes y 10 años de retención de datos en ausencia de energía. Ref. DS1230Y.
- Reloj de tiempo real de alta precisión, con batería auxiliar y manejo de hora y calendario. Ref. DS1307
- Microcontrolador de 8 bits, 40 pines y 8Kbytes de memoria Flash. Ref. PIC16F877
- Termómetro digital (sonda de temperatura). Ref. DS1620.
- Conversor de niveles de voltaje TTL-RS232.

Ventajas de la tarjeta MicroPublik:

- Ajuste de hora y programación de mensajes sencilla, vía transmisión serial RS232
- Software de programación fácil de usar y estandarizado para programar las tarjetas MicroPublik.

- Visualización de temperatura, hora (2 formatos) y fecha (3 formatos).
- Visualización de texto normal o negrilla.
- 5 tipos de efectos, en la visualización de mensajes: Rotate, Hold, Flash, Scroll, WipeUp.
- 32Kbytes de memoria disponible, para la programación de los mensajes.

PRINCIPIO DE FUNCIONAMIENTO

La tarjeta cuenta con varios componentes, los cuales cumplen funciones diferentes. Estos componentes se enumeran a continuación:

Microcontrolador Ref. PIC16F877

Este es un microcontrolador de 8 bits, 40 pines y 8Kbytes de memoria Flash, que se encarga de procesar toda la información necesaria, proveniente de la Memoria RAM, La sonda de temperatura, el reloj de tiempo real y el integrado de comunicación, para visualizar los mensajes en el informador.

La salida de datos a la etapa de potencia se maneja a través de un buffer 7416, que convierte los niveles lógicos TTL del microcontrolador, a niveles CMOS 0 – 12 voltios. Estos datos, están representados por las señales STRB, CLK, OE y DATA.

Memoria SRAM Ref. DS1230Y

Es la memoria donde queda guardada todos los mensajes que se visualizan en el informador. Esta memoria es una memoria de datos no volátil (no se borra en ausencia de energía), con una capacidad de almacenamiento de 32 Kbytes (32,768 caracteres) y 10 años de retención de datos en ausencia de energía. Esta memoria cuenta con 15 pines de direccionamiento (seleccionan una de las 32,768 posiciones de memoria), 8 pines de datos, 3 pines de control y 2 de alimentación. Para el direccionamiento de la ram se utilizan 2 registros de desplazamiento de 8 bits (4094), para reducir el número de pines necesarios en el microcontrolador. La lectura de datos se hace directamente desde la memoria hasta el microcontrolador.

Los registros de desplazamiento 4094, cumplen la función de convertir un byte recibido por la entrada de datos, a ocho salidas que representan el valor en binario del byte recibido. Si se conectan n registros 4094 en cascada, funcionan como si fuera un solo registro de $8 \times n$ salidas.

Reloj de tiempo real Ref. DS1307

En un circuito integrado IC de 8 pines, el cual cumple la función de llevar la noción del tiempo en años, meses, semanas, horas, minutos y segundos y va conectado directamente al microcontrolador. Este IC debe contar con una batería auxiliar, que le permite continuar con su funcionamiento normal en ausencia de energía eléctrica. Cabe destacar, que si la batería está descargada o no está conectada al integrado, éste no funciona correctamente aún si éste cuenta con el suministro normal de energía.

Termómetro digital (sonda de temperatura). Ref. DS1620.

Este es un integrado de 8 pines, encargado de enviar la información de temperatura, al microcontrolador.

Conversor de niveles de voltaje TTL-RS232.

Es el encargado de convertir los niveles de voltaje RS232, que maneja el puerto serial del computador, a los niveles TTL que soporta el microcontrolador, para así poderle enviar datos desde el computador.

INSTALACIÓN

Implementos necesarios:

- 1- Tarjeta Micropublik
- 2- Cable de datos A9
- 3- Adaptador de 12 Vdc
- 4- Avantel (si se va a realizar transmisión via Avantel)
- 5- Portátil (si la transmisión no se va a realizar por avantel)

Antes de alimentar la tarjeta con el adaptador, se deben conectar los cables de datos a la etapa de potencia del informador. Estos son:

SEÑALES TARJETA	ETAPA POTENCIA
GND	Negro
OE	Naranja
STRB	Verde
CLK	Rojo
DATA	Amarillo

Estas señales salen del conector Azul de tornillo 6 puestos de la tarjeta y entran a la etapa de potencia del informador.

También se debe verificar que esté conectada la sonda de temperatura en su respectivo conector blanco de 5 pines, el cual se encuentra dividido en dos conectores, uno de 3 pines y otro de 2 pines, uno invertido con respecto del otro.

Teniendo listos los anteriores puntos, ya se puede alimentar la tarjeta con el adaptador.

Programación de Mensajes:

La programación de la tarjeta, ya sea mensajes o otras configuraciones, se realiza por transmisión RS485 o por transmisión RS232. esta se puede hacer desde un Avantel o desde un computador con puerto serial.

Para cualquiera de las dos formas de transmisión (Avantel/Computador) la configuración del cable de datos para RS232 es:

Hembra (PC):

Pin	Función/Color
3	TX – Verde
2	RX - Naranja
5	Gnd - Blanco – x

a)

Macho (Tarjeta):

Pin	Función/Color
3	RX – Verde
2	TX - Naranja
5	Gnd - Blanco – x

b)

Tablas a y b. Configuración del cable A9 - uno a uno.

Por último, energice las etapas de control y de potencia y verifique el correcto funcionamiento de la tarjeta, teniendo en cuenta los siguientes aspectos:

- 1- La hora debe estar bien ajustada.
- 2- La programación de los mensajes debe ser la correcta.
- 3- La temperatura mostrada debe ser coherente
- 4- Todas las filas del informador deben estar funcionando bien.
- 5- Los efectos distintos al Rotate deben aparecer centrados en el informador.

Si fallan los puntos 1 y/o 5, se debe de transmitir a la tarjeta la hora o la configuración de verticales, desde el software, según sea el caso.

Si los problemas persisten, remítase al manual técnico de la tarjeta.

Notas adicionales:

Se debe tener precaución en ajustar la hora del computador con hora atómica (hora de GPS), antes de transmitir la hora a la tarjeta

El número mínimo de columnas necesarias en el informador, para que la hora, la temperatura y los decibels (dB) salgan bien en su totalidad en el efecto de hold, es de 26. Para informadores con una cantidad menor de columnas, se tendría que modificar el Firmware de la tarjeta con un tipo de fuente especial (más delgada), que permita visualizar esta información.

SOLUCIÓN DE PROBLEMAS

1. El informador no recibe información.

Revise la configuración del puerto en su computador. Verifique el transistor T2 2N3906 en la tarjeta, los diodos D6 y D7 y el integrado U7(max232)

2. No se puede visualizar nada en el display.

Verifique las señales DATA, CLK, STRB, OE en el circuito integrado 7416 .

Verifique las señales de los integrados (4094) que direccionan los datos de la Ram.

Verifique el integrado IC3 (DS1230 memoria RAM)

3. La hora se desactualiza ó muestra 00:00.

Verifique el estado de la batería de 3V y su la base y la continuidad en el integrado DS1307.

4. La hora muestra signos ? ej. 0?:??

Verifique continuidad en las líneas que van desde el DS1307 hasta el microcontrolador. Si las conexiones se muestran bien, reemplace el DS1307. Si eso no funciona, reemplace también el Microcontrolador.

5. La temperatura es errónea.

Verifique las conexiones de la sonda y reemplace el integrado de la sonda temperatura DS1620.

Si la temperatura visualizada se muestra como :1°C, el problema es de la sonda, o no hay continuidad entre el microcontrolador y el CI DS1620.

6. Sólo se visualiza un solo carácter en el informador o los caracteres que se visualizan son extraños.

Verifique las señales de los integrados IC4 e IC5 (4094) que direccionan los datos de la Ram.

Verifique el integrado(DS1230 memoria RAM)

REIVINDICACIÓN 1

OBJETO

La tarjeta MicroPublik es una tarjeta capaz de transmitir mensajes en un display electrónico, cuya etapa de potencia esté constituida por las tarjetas o las Board Driver de los informadores o equipos incandescentes. Esta tarjeta permite visualizar mensajes con cierta variedad de efectos que son: Rotate, Hold, Flash, Scroll, WipeUp. Estos efectos pueden ser programados con 10 velocidades distintas de visualización.

Esta tarjeta cuenta con otras funciones, como visualización de temperatura, hora en 2 formatos, fecha en tres formatos y texto en negrilla. La programación de los mensajes y las funciones mencionadas, se realizan vía puerto serial RS485 ó RS232 desde un software de programación

A continuación listamos los elementos de la Tarjeta MicroPublik, según los nombres dispuestos en el diseño (Esquemático e Impreso):

- H1, H2, H3, H4: Hoyos de fijación para la Tarjeta MicroPublik.
- P\$1, P\$2, P\$3: Conector de Entrada DC Jack.
- X2: Bornera de 2 puestos.
- VR1: Regulador de Voltaje de 12 Voltios.
- VR2: Regulador de Voltaje de 5 Voltios.
- D3, D4: Diodos de señal 1N4004.
- C22: Condensador electrolítico de 2200uF a 25V.
- C21: Condensador electrolítico de 1000uF a 16V.
- C17, C13, C19, C14, C16, C15, C11: Condensador de cerámica 104 (0,1uF o 100nF).
- C8: Condensador electrolítico de 470uF a 16V.
- C20: Condensador de tantalio de 1uF.
- U8: Circuito Integrado 7416, el cual tiene 4 compuertas negadoras.
- R13, R14, R15, R16: Resistencia de 470 Ohm a 1/4W.
- X1: Bornera de 6 puestos.
- RS485: Bornera de 3 puestos.
- U6: Circuito Integrado 75176: Es un circuito integrado diseñado para la comunicación bidireccional en líneas de transmisión multipunto. Está diseñado para líneas de transmisión balanceadas bajo el estándar EIA/TIA-422-B. Este chip posibilita la conversión entre los niveles TTL del micro-controlador y las señales del tipo diferencial que se utilizan el bus RS-485. Este integrado tiene dos líneas de control que dan 3 funciones diferentes al chip: transmisor, receptor y generador de alta impedancia, evitando así conflictos en el bus de datos.
- U\$1: Conector RJ-22: Sirve para las transmisiones de datos a través de RS485.
- R4, R5: Resistencias de 3.3KOhm a 1/4W.
- D6, D7, D8, D9, D10, D11, D12, D13, D14, D15: Diodos de señal 1N4148.
- C3, C4, C5, C6: Condensadores electrolíticos de 1uF a 25V.
- U\$2: Conector serial DB-9 hembra.
- U7: Circuito Integrado Max232: Es un conversor de niveles de voltaje TTL-RS232.
- C18, C7, C12: Condensador cerámico 103 (0,01uF o 10nF).
- T2, T3: Transistor 2N3906.
- R6: Resistencia de 330 Ohm a 1/4W.
- DL1, DL2, DL3: Diodos Led de encendido, de recepción de datos y envío respectivamente.
- R8, R10: Resistencia de 220 Ohm a 1/4W.
- R7, R9: Resistencia de 2.7 KOhm a 1/4W.
- C1, C2: Condensadores Ceramicos de 22pF.
- R11: Resistencia de 10 KOhm a 1/4W.
- R12: Resistencia de 1.5 KOhm a 1/4W.
- Q1, Q2: Cristal Oscilador de 32.768 KHz.
- R2, R3: Resistencia de 1 KOhm a 1/4W.
- R1: Resistencia de 6.8 KOhm a 1/4W.
- SL1, SL2: Conectores para el sensor de temperatura DS1620: Este es un integrado de 8 pines, encargado de enviar la información de temperatura, al microcontrolador.
- U2: Circuito Integrado DS1307: Es un circuito integrado IC de 8 pines, el cual cumple la función de llevar la noción del tiempo en años, meses, semanas, horas, minutos y segundos y va conectado directamente al microcontrolador. Este IC debe contar con una batería auxiliar (B1), que le permite continuar con su funcionamiento normal en ausencia de energía eléctrica. Cabe destacar, que si la batería está descargada o no está conectada al integrado, éste no funciona correctamente aún si éste cuenta con el suministro normal de energía.
- G2: Pila de 3V.
- IC7: Microcontrolador Pic16f877A.

- IC4, IC5: Circuito Integrado 4094: Registro de desplazamiento de 8 bits, para mover los datos de la Memoria SRam DS1230Y a la etapa de potencia del microcontrolador.
- IC3: Memoria SRam DS1230Y: Es una memoria no volátil donde se almacenan los datos a mostrar.

Dado el diseño con tantos componentes, se divide este en 5 hojas, cada hoja del diseño corresponde a una etapa del producto. En el caso del diseño del impreso, se muestran diferentes capas del mismo para ver como quedara finalmente.

- Diagrama 1. Esquemático de la Tarjeta MicroPublik - Etapa de Control
- Diagrama 2. Esquemático de la Tarjeta MicroPublik - Etapa de Comunicación
- Diagrama 3. Esquemático de la Tarjeta MicroPublik - Etapa a la Salida de Potencia
- Diagrama 4. Esquemático de la Tarjeta MicroPublik - Fuentes de Alimentación
- Diagrama 5. Esquemático de la Tarjeta MicroPublik - Hoyos para fijación de Tarjeta
- Grafica 1. Impreso de la Tarjeta MicroPublik - Componentes
- Grafica 2. Impreso de la Tarjeta MicroPublik - Cara Superior
- Grafica 3. Impreso de la Tarjeta MicroPublik - Cara Inferior
- Grafica 4. Impreso de la Tarjeta MicroPublik - Componentes y ambas caras
- Grafica 5. Tarjeta MicroPublik v2.0

Código Fuente de MicroPublik en Asembler:

```
list p=16F877A
include<p16f877A.inc>
include <tabla_car_norm.h>
;Area FSR reservada para alamacenar datos temporales para escritura/lectura en EEPROM de datos y
Reloj Serial:
;20h - 2Fh
;Definicion de registros de propósito general (Variables) en Banco 0
cblock 0x28
;está funcionando bien, lellendo mensajes desde la flash del micro, sin la ram
;ajustar el scroll ;STATUS
;IRP RP1 RP0 TO# PD# Z DC C
RETARD ;PARA USO GENERAL DE LOS RETARDOS
COLMNASH ;PARA CONTAR EL NUMERO DE COLUMNAS ENVIADAS AL INFORMADOR EN LA
FRASE
COLMNASL
COLMNASHH ;PARA EL CONTEO DE LAS COLUMNAS DE ESPACIOS EN BLANCO
COLMNASLL
TPO_DESP
tiempo_desp
RETA_FLASH ;retardos
RETA_HOLD
RET_SCRL
RET_WIPE
C_RETA_EFEC ;contadores de retardos
ROTA_SCROLL
ROTA_WIPE_UP
COL_SCROLL
COL_WIPE
COL_AUX ;VARIABLE AUXILIAR PARA GUARDAR EL ESTADO DE LAS COLUMNAS
CARAUX2 ;CARACTER AUXILIAR PARA EL MANEJO DE LAS TABLAS
EFFECT_AUX ;VARIABLE AUXILIAR PARA CADA EFECTO
VECES ;NUMERO DE VECES QUE EJECUTA CADA EFECTO
NCOLUMNAS_H ;CONTADOR DE COLUMNAS
NCOLUMNAS_L
bandera0
bandera1
bandeMod
bandera2
```

contcolenvL
contcolenvH
contbits
DATOCOL
DIR_RAM_H
caracter
DIRH
DIRL
datox
DirIniH
DirIniL
NCVH
NCVL
NCTH
NCTL
cursor
cursor_ini
NCSEP
an_o
NCSEPAux
DESPLAZL
DESPLAZH
caraux
PCLATHtemp
DespOffsetL
DespOffsetH
cuenta
DIRECCIONR
RECEP
EEPROM
TRANSM
DATOPC
mes
dia_mes
dia_sem
hora
minutos
DIRVERHR_H
DIRVERHR_L
DIRVERFE_H
DIRVERFE_L
contesp
codascii
CODE_INFO
cuantos
cont_seg
segundos
posic_car
CUATRO ;77
Wtemp
DIR_RAM_HL
DIRH_HT
DIRL_HT
rx_retardl
rx_retardh
rx_retardhh
temp_l
temp_h
STATUS_TEMP

endc

cblock 0xA0

DIRH_TMP_b2
DIRL_TMP_b2
DATORX_b2

```

DECENA_b2
UNIDAD_b2
DIRVERTMP_H_b2
dirvertmp_l_b2
conta4_b2
datotx_b2
dato_b2
endc
;Señales de control de los registros de desplazamiento (4094) del PANEL:
#define DATAS PORTA,0 ;TRAMA DE BITS DE DATOS
#define CLOCK PORTA,1 ;ÉSTE, CON UN FLANCO DE SUBIDA LEE U NUEVO BIT DE LA
TRAMA DE BITS DE DATOS DE LA ENTRADA
#define STROBE PORTA,2 ;CARGA EN EL 4094 LO QUE HAY EN SU REGITRO DE ENTRADA
#define OE PORTC,2 ;HABILITA A LA SALIDA DEL 4094 EL DATO CARGADO EN ÉL (PWM)
;Señales de control de los registros de desplazamiento (4094) para enviar Dirección de la RAM
(DIRH_DIRL):
#define STR94 PORTD,7
#define DAT94 PORTD,6
#define CLK94 PORTD,5
#define OE94 PORTD,4
;Señales de control de memoria RAM:
#define WE PORTD,1 ;Selección Lectura/Escritura de memoria RAM
#define OER PORTD,2 ;Habilita búfer de salida de datos de memoria RAM
#define CS PORTD,3 ;Selección (habilitación) de memoria RAM
;Señal de control para habilitar/deshabilitar transmisión serial al PC [RS485]:
#define HABTX PORTC,5
;Señales de control del Reloj de Tiempo Real DS1307:
#define SCL PORTC,3
#define SDA PORTC,4
;Señales de control del termómetro digital DS1620
#define RSTT PORTC,0 ;pin 3 ds1620 pin4 gnd
#define CLKT PORTC,1 ;pin 2 pin8 vdd
#define DQT PORTD,0 ;pin 1
#define TRIS_DQT TRISD,0
;Bits del registro EEPROM asociados con el bus I2C
BIT_IN EQU 7
BIT_OUT EQU 6
#define fuente2 bandera2,0
#define CODEVELO BANDERA2,1
#define HORA_AMPM BANDERA2,2
#define hora_militar bandera2,3
#define sirx_retard BANDERA2,4
#define TEMPE_HR_FE BANDERA2,5
#define fin_digit bandera2,6
;#define wide_hold bandera2,7
#define one_second bandera1,0 ;indica que ha transcurrido 1 segundo
#define time_desp bandera1,1 ;Indica si se ha cumplido tiempo de desplazamiento horizontal.
#define fuente bandera1,2 ;Bandera modo de texto: 0 para NORMAL y 1 para NEGRITA.
#define fuente_ini bandera1,3 ;Indica estado fuente de texto al iniciar el refresco
#define fin_car bandera1,4 ;Fin de caracter enviado [sigue separación en blanco].
#define fin_car_ini bandera1,5 ;Fin de caracter inicial [sigue siguiente caracter]
#define aborto bandera1,6 ;Indica si se completó Número de Columnas Total del Panel
#define colvacias bandera1,7 ;Indica si existen columnas vacías o no en la rutina de refresco.
;#DEFINE modowide BANDEMODO,0
#define modoRotate bandeMod,1 ;Modo sistema: ROTATE.
#define modoRotate2 bandeMod,2 ;Modo sistema:
#define modohold bandeMod,3 ;Modo sistema:
#define modowipe_up bandeMod,4 ;Modo sistema:
#define modoscroll bandeMod,5 ;Modo sistema:
#define modoflash bandeMod,6 ;Modo sistema:
#define MODOHOLDW BANDEMODO,7 ;
#define fuente_fe bandera0,0 ;iniciador de negrilla en los efectos
#define fuente_fe2 bandera0,1 ;iniciador de negrilla 2
#define FUENTE_INI2 bandera0,2
#define FECHA01 bandera0,3 ;SELECTOR DE FORMATO DE FECHA 1
#define FECHA02 bandera0,4 ;SELECTOR DE FORMATO DE FECHA 2
#define FECHACODE bandera0,5 ;bandera de codigo de fecha
#define continuarr bandera0,6 ;bandera indicadores de continuar en el ciclo de inicializaicon

```

[illegible]

```

addwf PCL,1
dt "Ene"
dt 0x7B ;espacio corto
dt "Feb"
dt 0x7B
dt "Mar"
dt 0x7B
dt "Abr"
dt 0x7B
dt "May"
dt 0x7B
dt "Jun"
dt 0x7B
dt "Jul"
dt 0x7B
dt "Ago"
dt 0x7B
dt "Sep"
dt 0x7B
dt "Oct"
dt 0x7B
dt "Nov"
dt 0x7B
dt "Dic"
dt 0x7B
IntSerRout
    MOVWF WTEMP
    SWAPF STATUS,W
    CLRF STATUS
    MOVWF STATUS_TEMP
    MOVF PCLATH,W
    MOVWF PCLATHTEMP
; bcf INTCON,PEIE
; bcf intcon,gie
    CLRF INTCON ;Deshabilita todas las interrupciones
    btfss PIR1,RCIF
    goto NO_INTSER ;NO HAY INTERRUPCIÓN VÁLIDA
    clrwdt
    bsf pclath,3
    bsf pclath,2
    call RutinaINT
    clrwdt
NO_INTSER
; bcf STATUS,RP0
; BCF ONE_SECOND
; BCF PIR1,TMR1IF
    bsf STATUS,RP0
    bsf PIE1,RCIE ;Habilita interrupción para recepción de USART
    BSF TXSTA,TXEN ;HABILITA TRANSMISIÓN
    bcf STATUS,RP0
    bsf RCSTA,SPEN
    bsf RCSTA,CREN ;Habilita recepción continua.
    bcf RCSTA,RX9 ;Dato de recepción de 8 bits.
    clrf intcon ;dashabilita interrupciones y algunos flags
    clrf PIR1 ;borra los demás flags de interrupcion
;SWAPF NO AFECTA EL REGISTRO STATUS
    MOVF PCLATHTEMP, W ;Restore PCLATH
    MOVWF PCLATH ;Move W into PCLATH
    SWAPF STATUS_TEMP,W ;Swap STATUS_TEMP register into W
    MOVWF STATUS ;Move W into STATUS register
    SWAPF WTEMP,F ;Swap W_TEMP
    SWAPF WTEMP,W ;Swap W_TEMP into W
    retfie
INICIO
    bsf pclath,3
    bsf pclath,2
    CALL INICIALIZACION
    CLRF PCLATH

```

```

    btfsc continuarr
    goto continuar
;Ahora busca dirección RAM donde se encuentra el primer caracter ascii para empezar [DirIni]:
call BUSCADIRINI
; call SETUPRAM ;No se encontró código de fin de mensaje, entonces graba mensaje por defecto.
CONTINUAR
    BSF MODOROTATE ;MODO ROTATE POR DEFECTO
    clrf TMR1H
    clrf TMR1L
    bsf T1CON,TMR1ON
    bsf INTCON,PEIE
    bsf INTCON,GIE ;Habilita todas las interrupciones
;Configurar USART para comunicación serial generando interrupción.
    clrf RCSTA
    bsf RCSTA,SPEN ;Configura RC7(RX) y RC6(TX) como pines del puerto serial.
    bsf STATUS,RP0
    bsf PIE1,RCIE ;Habilita interrupción para recepción de USART
    bcf TXSTA,SYNC ;Selecciona modo ASINCRONO de la USART.
    bcf TXSTA,TX9 ;Dato de transmisión de 8 bits.
    bsf TXSTA,BRGH ;Escoge velocidad de comunicación ALTA.
    BSF TXSTA,TXEN ;HABILITA TRANSMISIÓN
    movlw .129 ;Establece velocidad de comunicación (baudios).
    movwf SPBRG ;Para un cristal de 20 MHz.
; movlw .77 ;Establece velocidad de comunicación (baudios).
; movwf SPBRG ;Para un cristal de 12 MHz.PRUEBABA
    bcf STATUS,RP0
    bsf RCSTA,CREN ;Habilita recepción continua.
    bcf RCSTA,RX9 ;Dato de recepción de 8 bits.
; ***** prueba trasmisión
; movlw 0x31
; bcf PIR1,TXIF ;Restaura el flag del transmisor
; movwf TXREG ;Almacena el byte a transmitir
; bsf STATUS,RP0
;Tx_Dato_Waitx1
; btfss TXSTA,TRMT ;Byte transmitido ??
; goto Tx_Dato_Waitx1 ;No, esperar
; bcf STATUS,RP0
; movlw 0x32
; bcf PIR1,TXIF ;Restaura el flag del transmisor
; movwf TXREG ;Almacena el byte a transmitir
; bsf STATUS,RP0
;Tx_Dato_Waitx2
; btfss TXSTA,TRMT ;Byte transmitido ??
; goto Tx_Dato_Waitx2 ;No, esperar
; bcf STATUS,RP0
; bsf pclath,3
; bsf pclath,2
; call respuesta
; CLRF PCLATH
; *****espera de dos segundos
espera clrwdt
    MOVLW 0xFF
    XORWF TMR1H,W
    BTFSS STATUS,Z
    goto espera
esp_ov ;ESPERA EL OVERFLOW
    CLRWDT
    MOVLW 0x00
    XORWF TMR1H,W
    BTFSS STATUS,Z
    goto esp_ov
    incf cont_seg,1
    movlw .76
    xorwf cont_seg,W
    btfss STATUS,Z
    goto espera
    clrf cont_seg
;Uso del PWM

```

22
24

```

; bsf STATUS,RP0 ;switch bank 1
; movlw periodo ; cargamos el valor del periodo en PR2
; movwf PR2
; bcf STATUS,RP0 ;switch bank 0
; movlw duty ; cargamos valor del duty en CCP1L
; movwf CCP1L
pwmon
; bsf T2CON,2 ;TMR2 activado
; movlw 0x0c ;cargamos setup módulo
; movwf CCP1CON ;PWM
; Estas lineas para desactivar la salida
;pwmoﬀ clrf CCP1CON
; bcf T2CON,2 ;Desactivamos TMR2
; bcf PWM ;esta línea en caso que al finalizar la emisión el pin quede a 1.
BSF OE
goto ciclo
;-----
ACTUALZR_HT
CALL ACTUALIZA_H_T
BCF TEMPE_HR_FE
;*****
;*****
;*****
;*****
;*****
CICLO
BTFSC TEMPE_HR_FE
GOTO ACTUALZR_HT
; movlw duty ; recargamos valor del duty en CCP1L
; movwf CCP1L
movf tmr1h,w ;control del timer overflow
addlw .80
movwf tmr1h
WIT_OVRFW0 ;ESPERA EL OVERFLOW
clrwdt
MOVLW 0xFF
XORWF TMR1H,W
BTFSS STATUS,Z
goto WIT_OVRFW0
WIT_OVRFW
CLRWDT
MOVLW 0x00
XORWF TMR1H,W
BTFSS STATUS,Z
goto WIT_OVRFW
;*****
;*****
;*****
TIMER_OVERFLOW
bsf INTCON,PEIE
bsf INTCON,GIE ;Habilita todas las interrupciones
clrwdt
bcf status,z
btfsc modorotate2
bsf modorotate
;Se pregunta si está en modo Rotate ???
incf tiempo_desp,1
movf TPO_DESP,W
xorwf tiempo_desp,W
btfss STATUS,Z
goto CHSEG
bsf time_desp ;Cumplido tiempo de desplazamiento.
clrf tiempo_desp
CHSEG
incf cont_seg,1
movlw .38 ;OJO, con preescal de Timer 1 en 1:2 (tpo_rebose = 13,1 ms). Entonces con preescala de
1:2,
xorwf cont_seg,W ;la interrupción ocurre cada: 2*tpo_rebose = 26,2 ms. Para un tiempo transcurrido de
un segundo,

```

23
22

```

btfss STATUS,Z ;el número de veces que se debe detectar la interrupciones de 38 ! [38*26.2ms =
997ms]
goto SIGREF_XX
;Aquí se lee la hora/fecha y se almacena en RAM si existe el atributo:
BSF ONE_SECOND
clrf cont_seg
bcf status,z
incf segundos,1
movlw .15
xorwf segundos,w
btfss status,z
goto SIGREF_XX
clrf segundos
bsf INTCON,PEIE
bsf INTCON,GIE ;Habilita todas las interrupciones
;Configurar USART para comunicación serial generando interrupción.
clrf RCSTA
bsf RCSTA,SPEN ;Configura RC7(RX) y RC6(TX) como pines del puerto serial.
bsf STATUS,RP0
bsf PIE1,RCIE ;Habilita interrupción para recepción de USART
bcf TXSTA,SYNC ;Selecciona modo ASINCRONO de la USART.
bcf TXSTA,TX9 ;Dato de transmisión de 8 bits.
bsf TXSTA,BRGH ;Escoge velocidad de comunicación ALTA.
BSF TXSTA,TXEN
movlw .129 ;Establece velocidad de comunicación (baudios).
movwf SPBRG ;Para un cristal de 20 MHz.
; movlw .77 ;Establece velocidad de comunicación (baudios).
; movwf SPBRG ;Para un cristal de 12 MHz.PRUEBABA
bcf STATUS,RP0
bsf RCSTA,CREN ;Habilita recepción continua.
bcf RCSTA,RX9 ;Dato de recepción de 8 bits.
bcf status,rp1
btfss PIR1,RCIF ;PARA DESCARGAR EL BUFFER
GOTO SIGREF_XX
bsf pclath,3
bsf pclath,2
call RutinaINT
; bsf STATUS,RP0 ;HABILITA RX SERIAL POR SILAS
; bsf PIE1,RCIE
; bcf STATUS,RP0
SIGREF_XX
; btfss modoRotate ;El sistema se encuentra en modo ROTATE?
; goto sigref ;No
; btfsc time_desp ;Cumplido tiempo de desplazamiento?
; goto SIGREF ;Si.
NOVRFLW ;ESPERA PARA QUE SE REINICIE EL TIMER
clrwdt
MOVLW 0xFF
XORWF TMR1H,W
BTFSC STATUS,Z
goto NOVRFLW ;No. SE HA COMPLETADO EL TIEMPO DE ESPERA DEL TIMER
; goto FIN_TIM
SIGREF
call REFRESCA
; bcf PIR1,TMR1IF
btfss modoRotate ;El sistema se encuentra en modo ROTATE?
goto FIN_TIM ;No
MODROT
clrwdt
btfss time_desp ;Cumplido tiempo de desplazamiento?
goto FIN_TIM ;No.
bcf time_desp
btfsc colvacias ;Existen columnas vacías? pruebaba debe ser activado
goto INCCURSORINI ;NO!, entonces se incrementa valor del cursor inicial (incluye separación de
caracteres)
;Si, entonces decrementa el número de columnas vacías:
clrwdt
bcf STATUS,C

```

```

movlw .1
subwf NCVL,1
btfss STATUS,Z
goto MIRANEG
movlw .0
xorwf NCVH,W
btfss STATUS,Z
goto FIN_TIM
; clrf cursor_ini
bsf colvacias ;Ya no hay columnas vacías!!
; call BUSCADIRINI
; CLRF CURSOR_INI
goto FIN_TIM
MIRANEG
btfsc STATUS,C ;resultado de resta anterior negativo?
goto FIN_TIM ;No!
decf NCVH,1
goto FIN_TIM
INCCURSORINI ;cursor ini es la posicio inicial de la columna del caracter inicial.
;Al final de un caracter inicial, se incrementa Dirección Inicial RAM y Cursor Inicial = 0.
;Lee dato de columna en posición cursor_ini de caracter inicial [en DirIniH_DirIniL] para
;determinar si es fin de caracter inicial:
clrwdt
movf DirIniH,W
movwf DIRH
movf DirIniL,W
movwf DIRL
call LEE_RAM
call calcula_offset ;CALCULA LA POSICION DEL CARACTER EN LA TABLA DE CARACTERES QUE
ESTÁ AL FINAL
movf DespOffsetL,W ;Y QUEDA GUARDADA EN DespOffsetH y DespOffsetL
movwf DESPLAZL
movf DespOffsetH,W
movwf DESPLAZH
movf cursor_ini,W
addwf DESPLAZL,1
btfsc STATUS,C
incf DESPLAZH,1
bcf fuente
btfsc fuente_ini
bsf fuente
call BUSCACOL
;Chequea si es fin de caracter [separación] para incrementar Dirección Inicial de RAM
btfsc DATOCOL,0
goto FINCARINI ;Sí es fin de caracter inicial.
incf cursor_ini,1 ;No es fin de caracter inicial.
goto FIN_TIM
FINCARINI
clrwdt
clrf cursor_ini
incf DirIniL,1
btfsc STATUS,Z
incf DirIniH,1
MARCAFMSGE
movf DirIniH,W
movwf DIRH
movwf DIRH_HT
movf DirIniL,W
movwf DIRL
movwf DIRL_HT
call LEE_RAM
movlw CODE_TNORM ;Código de texto NORMAL
xorwf caracter,W
btfss STATUS,Z
goto salxx1
CALL code_tnorm_cod
goto FINCARINI
salxx1

```

25
24

```
    clrwdt
    movlw CODE_TNEG    ;Código de texto NEGRITA
    xorwf character,W
    btfss STATUS,Z
    goto SALXX1X
    call code_tneg_cod
    goto FINCARINI
SALXX1X
    movlw CODE_TNEG2    ;Código de texto NEGRITA2
    xorwf character,W
    btfss STATUS,Z
    goto salxx2
    call code_tneg2_cod
    goto FINCARINI
salxx2
    clrwdt
    movlw CODE_HORA1    ;Código de HORA1
    xorwf character,W
    btfss STATUS,Z
    goto salxx3
    call code_hora1_cod
    goto FINCARINI
salxx3
    clrwdt
    movlw CODE_HORA2    ;Código para mostrar la HORA2
    xorwf character,W
    btfss STATUS,Z
    goto salxx4
    call code_hora2_cod
    goto FINCARINI
salxx4
    clrwdt
    CALL BUSCA_FECHACODE
    BTFSC FECHACODE
    goto FINCARINI
salxxx4
    movlw temperatura
    xorwf character,W
    btfss STATUS,Z
    goto salxx5
    call temperatura_cod
    goto FINCARINI
salxx5
    movlw CODE_FINEFECTO ;Código para FIN de mensajes!
    xorwf character,W
    btfss STATUS,Z
    goto salxx6
    goto FINCARINI
salxx6
    clrwdt
    movlw CAR_ROTATE
    xorwf character,W
    btfss STATUS,Z
    goto salxx611
    call car_rotate_cod
    goto fincarini
salxx611
    call code_velo    ;codigo de velocidad
    btfss codevelo
    GOTO salxx7
    CALL set_codevelo_rot
    goto fincarini
salxx7
    clrwdt
    movlw CODE_FINMENSAJE ;Código para FIN de mensajes!
    xorwf character,W
    btfss STATUS,Z
    goto FIN_TIM    ;Es código ASCII de caracter!
```

FINALMSG

```
    clrwdt      ;FIN de mensajes!
```

```
    call  BUSCADIRINI
```

```
    ;Establece número de columnas vacías para separación de mensajes:
```

```
    clrf  tiempo_desp
```

FIN_TIM

```
    bsf   STATUS,RP0
```

```
    bsf   PIE1,RCIE
```

```
    bcf   STATUS,RP0
```

```
    BCF   ONE_SECOND
```

```
    goto  CICLO
```

```
*****
```

```
*****
```

```
*****
```

```
    ; AQUÍ SE ENVÍA TODA LA TRAMA COMPLETA PARA LLENAR EL INFORMADOR, ES DECIR, SE
```

```
    ;ACTUALIZA.
```

```
*****
```

```
*****
```

```
*****
```

REFRESCA

```
    ;
```

```
    clrwdt
```

```
    clrf  contcolenvL    ;contador columnas enviadas
```

```
    clrf  contcolenvH
```

```
    bcf   STROBE
```

```
    bcf   status,z
```

```
    movlw .0
```

```
    xorwf bandemod,w
```

```
    BTFSC status,z
```

```
    GOTO  SIGA_ROTAR    ;
```

```
    btfss modorotate    ;El sistema se encuentra en modo ROTATE?
```

```
    GOTO  SIGARAM    ;si no hay modo rotate activado salte a la lectura de la ram
```

siga_rotar

```
    btfsc colvacias    ;Existen columnas vacías para enviar inicialmente?
```

```
    goto  SIGARAM    ;NO!
```

ENVCOLVAC

```
    clrwdt    ;SI!, todavía.
```

```
    clrf  DATOCOL
```

```
    call  ENVIA7
```

```
    incf  contcolenvL,1
```

```
    btfsc STATUS,Z
```

```
    incf  contcolenvH,1
```

```
    ; [Cont column enviadas] = [Número de columnas vacías existentes]?
```

```
    movf  NCVH,W
```

```
    xorwf contcolenvH,W
```

```
    btfss STATUS,Z
```

```
    goto  ENVCOLVAC
```

```
    movf  NCVL,W
```

```
    xorwf contcolenvL,W
```

```
    btfss STATUS,Z
```

```
    goto  ENVCOLVAC
```

```
    clrwdt
```

SIGARAM

```
    clrwdt
```

```
    movf  cursor_ini,W
```

```
    movwf cursor
```

```
    movf  DirIniH,W
```

```
    movwf DIRH
```

```
    movf  DirIniL,W
```

```
    movwf DIRL
```

```
    bcf   fuente
```

```
    btfsc fuente_ini
```

```
    bsf   fuente
```

CICLECT

```
    clrwdt
```

```
    call  LEE_RAM    ;Lee código de la posición de memoria DIRH_DIRL de RAM. El dato leído se  
    almacena en 'caracter'
```

```
    movf  dirl,w
```

```
    movwf dirl_ht
```

```
    movf  dirh,w
```

29
26

```
movwf dirh_ht
movlw code_tnorm
xorwf character,W
btfsc STATUS,Z
goto TEXTONORM ;Código de texto NORMAL
movlw code_tneg
xorwf character,W
btfsc STATUS,Z
goto TEXTONEGR ;Código de texto NEGRITA
movlw code_tneg2
xorwf character,W
btfsc STATUS,Z
goto TEXTONEGR2 ;Código de texto NEGRITA
movlw code_hora1
xorwf character,W
btfsc STATUS,Z
goto MOSTHORA ;Código para mostrar la HORA
movlw code_hora2
xorwf character,W
btfsc STATUS,Z
goto MOSTHORA2 ;Código para mostrar la HORA2
MOVLW TEMPERATURA
XORWF CHARACTER,W
btfsc STATUS,Z
GOTO COD_TEMPERATURA ;Código para TMP
CALL BUSCA_FECHACODE
BTFSC FECHACODE
GOTO CONTI_CICLECT
; movlw 0x09
; xorwf character,W
; btfsc STATUS,Z
; goto MOSTFECHA ;Código para mostrar la FECHA
;*****
;*****
;
MOVLW CAR_HOLD
xorwf character,W
btfsc STATUS,Z
goto EF_HOLD ;Código para EFECTO HOLD
MOVLW CAR_FLASH
xorwf character,W
btfsc STATUS,Z
goto EF_FLASH ;Código para EFECTO FLASH
MOVLW CAR_SCROLL
xorwf character,W
btfsc STATUS,Z
goto EF_SCROLL ;Código para EFECTO SCROLL
MOVLW CAR_WIPEUP
xorwf character,W
btfsc STATUS,Z
goto EF_WIPE_UP ;Código para EFECTO WIPE UP
; MOVLW CAR_WIDE
; xorwf character,W
; btfsc STATUS,Z
; goto EF_WIDE ;Código para EFECTO WIPE UP
MOVLW CAR_ROTATE
xorwf character,W
btfsc STATUS,Z
GOTO EF_ROTATE ;Código para EFECTO ROTATE
;*****
;*****
;
movlw code_finefecto ;Código para FIN de EFECTOS
xorwf character,W
btfsc STATUS,Z
GOTO CONTI_CICLECT
movlw code_finmensaje
xorwf character,W
btfsc STATUS,Z
```

```

goto FINALMSGXX ;Código para FIN de mensajes!
call code_velo ;busca codigo de velocidad
btfsc codevelo
goto inc_digi
GOTO MOSTCAR
INC_DIGI ;OTROS: 1 CAR_NORM; 3 CAR_NEG; 7 HORA; 6,8,9;FECHA
call set_codevelo_rot
CALL INC_DIRLH
GOTO CICLECT
;ENTONCES ES CÓDIGO ASCII DE CHARACTER!
;SE REINICIAN LOS REGISTROS QUE SE USAN EN LOS EFECTOS
MOSTCAR clrwdt ;MODO ROTATE
; movlw duty ; se incrementa el valor del duty en CCPR1L
; addlw .100 ;para mejorar la visualizacion del rotate
; MOVLW PERIODO ;PARA QUE EL DUTY SEA IGUAL AL PERIODO
; movwf CCPR1L
call ENVCOLCAR ;ENVIA COLUMNAS DEL CHARACTER CORRESPONDIENTE. SI SE ACABARON
LAS COLUMNAS => BSET aborto
btfsc aborto
goto SALIREF ;SE LLENÓ EL INFORMADOR (YA NO HAY COLUMNAS VACÍAS)
CALL INC_DIRLH
clrf cursor
goto CICLECT
CONTI_CICLECT
CALL INC_DIRLH
clrf cursor
goto CICLECT
;*****
FINALMSGXX
clrwdt
;Se envían espacios en blanco, es decir, la separación entre el mensaje y la repetición del mismo.
bcf fuente
bcf fuente_fe
movf NCSEP,W
movwf NCSEPAux
clrf DIRH
clrf DIRL
clrf cursor
BTFSC MODOROTATE
goto CICLECT
CALL BUSCADIRINI
BCF STATUS,Z
INCF NCVL,1
BTFSC STATUS,Z
INCF NCVH,1
RETURN
;*****
TEXTONORM
clrwdt
bcf fuente
bcf fuente_fe
BTFSS MODOROTATE
bcf fuente_ini
GOTO CONTI_CICLECT
;*****
TEXTONEGR
clrwdt
bsf fuente
bsf fuente_fe
GOTO CONTI_CICLECT
;*****
TEXTONEGR2
clrwdt
bsf fuente2
bsf fuente_fe2
GOTO CONTI_CICLECT
;*****
MOSTHORA

```

29
28

```
CALL CODE_HORA1_COD
GOTO CONTI_CICLECT ;No se ha terminado el refresco, entonces continúa leyendo datos de RAM!
;*****
MOSTHORA2
CALL CODE_HORA2_COD
GOTO CONTI_CICLECT ;No se ha terminado el refresco, entonces continúa leyendo datos de RAM!
;*****
COD_TEMPERATURA
CALL TEMPERATURA_COD
GOTO CONTI_CICLECT
;*****
EF_HOLD
MOVF DIRL,W ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
MOVWF DIRINIL
MOVF DIRH,W
MOVWF DIRINIH
BTFSC MODOHOLD
GOTO SIG_HOLD0
MOVF RETA_HOLD,W
MOVWF C_RETA_EFEC
SIG_HOLD0
clrwdt
CALL LIMPIA_PANEL ;LIMPIA EL PANEL
clrf cursor
CLRF BANDEMODO
BSF MODOHOLD
BCF STATUS,C ;BORRA EL BIT DE CARRY
CALL ENVIA_FRASE
BTFSC ONE_SECOND
DECFSZ C_RETA_EFEC,1 ;DECREMENTE Y SALTE SI CERO. DECREMENTA ERTA_HOLD Y SI
CERO, FIN_HOLD
GOTO SIG_HOLD
GOTO FIN_HOLD ;FIN DE CICLO HOLD
SIG_HOLD
NOP
NOP
NOP
BSF STROBE
RETURN
FIN_HOLD
GOTO FIN_EFECTS
;*****
EF_FLASH
clrwdt
MOVF DIRL,W ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
MOVWF DIRINIL
MOVF DIRH,W
MOVWF DIRINIH
BTFSC MODOFASH
GOTO SIG_FLASH0
MOVF RETA_FLASH,W
MOVWF C_RETA_EFEC
SIG_FLASH0
CLRF BANDEMODO
BSF MODOFASH ;MODO_FLASH
CALL LIMPIA_PANEL ;LIMPIA EL PANEL
BCF STATUS,Z
DECFSZ C_RETA_EFEC,1
GOTO FOLLOW_FLASH
CAMBIO_FLASH
BCF STATUS,Z
INCF EFECT_AUX,1
MOVF RETA_FLASH,W
MOVWF C_RETA_EFEC
FOLLOW_FLASH
```

30
29

```

    BTFSC  EFECT_AUX,0      ;TESTEA SI EL REGITRO CONTADOR DE VECES FLASH ES PAR O
IMPAR.
    GOTO  FLASH_IMPAR
FLASH_PAR
    clrf  cursor
    call  ENVIA_FRASE      ;ENVIA LA FRASE
    MOVLW .10              ;PARA CONTRARESTAR EL TIEMPO DE MAS QUE SE GENERA EN EL MODO
PAR
    CALL  RETA              ;CON LIMPIA PANEL
    goto  flas_ya
FLASH_IMPAR
    call  ENVIA_FRASE      ;ENVIA LA FRASE
    CALL  LIMPIA_PANEL     ;LIMPIA EL PANEL
    goto  flas_ya
flas_ya
    BSF  STROBE
    BCF  STATUS,Z
    MOVLW VECES_FLASH      ;CAMBIAR X CONSTANTE
    XORWF EFECT_AUX,W
    BTFSC STATUS,Z
    GOTO  FIN_FLASH
    RETURN
FIN_FLASH
;    MOVLW TIME_FLASH
;    MOVWF RETA_FLASH
    GOTO  FIN_EFFECTS
;*****
EF_SCROLL
    MOVF  DIRL,W           ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
    MOVWF DIRINIL
    MOVF  DIRH,W
    MOVWF DIRINIH
    BTFSC MODOSCROLL
    GOTO  SIG_SCROLL
    MOVF  RET_SCRL,W
    MOVWF C_RETA_EFEC
SIG_SCROLL
    clrwdt
    clrf  cursor
    CLRF  BANDEMODO
    BSF  MODOSCROLL
    CALL  LIMPIA_PANEL
SCRL
    BCF  STATUS,C          ;BORRA EL BIT DE CARRY
    BCF  STROBE
    CALL  ENVIA_FRASE
    MOVLW .90
    CALL  RETA
    BCF  STATUS,Z
    INCF  EFECT_AUX,1
    MOVLW TIME_SCROLL
    XORWF EFECT_AUX,W
    BTFSS STATUS,Z
    GOTO  SIGA_SCRLL
    CLRF  EFECT_AUX
    BCF  STATUS,Z
    MOVLW .9
    XORWF ROTA_SCROLL,W
    BTFSC STATUS,Z
    GOTO  SCRL_PAUSA
SCRL_SIGA2
    INCF  COL_SCROLL,1
    DECFSZ ROTA_SCROLL,1   ;VERIFICA SI YA SE REALIZARON TODOS LO CORRIMIENTOS
    GOTO  SIGA_SCRLL
    GOTO  FIN_SCROLL
SCRL_PAUSA
    DECFSZ C_RETA_EFEC,1

```

```
GOTO SIGA_SCRLL
GOTO SCRL_SIGA2
SIGA_SCRLL
BSF STROBE
RETURN
FIN_SCROLL
MOVLW .16
MOVWF ROTA_SCROLL
MOVLW .1
MOVWF COL_SCROLL ;GUARDA LA VARIABLE DE ROTACIÓN DE LAS COLUMNAS
GOTO FIN_EFECTS
*****
EF_WIPE_UP
MOVF DIRL,W ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
MOVWF DIRINIL
MOVF DIRH,W
MOVWF DIRINI
BTFSK MODOHOLDW
GOTO SIG_WIPEO
MOVF RET_WIPE,W
MOVWF C_RETA_EFEC
SIG_WIPEO
BTFSK MODOWIPE_UP
GOTO WIPE_MODD
BTFSK MODOHOLDW
GOTO FIN_WIPE
WIPE_MODD
clrwdt
clrf cursor
CLRF BANDEMODO
BSF MODOWIPE_UP
CALL LIMPIA_PANEL
WIPE_SCRLL
BCF STATUS,C ;BORRA EL BIT DE CARRY
BCF STROBE
CALL ENVIA_FRASE
MOVLW .90
CALL RETA
INCF EFECT_AUX,1
MOVLW TIME_WIPE
XORWF EFECT_AUX,W
BTFSK STATUS,Z
GOTO SIG_WIPE
INCF COL_WIPE,1
CLRF EFECT_AUX
DECFSZ ROTA_WIPE_UP,1 ;VERIFICA SI YA SE REALIZARON TODOS LOS CORRIMIENTOS
GOTO SIG_WIPE
GOTO FIN_WIPP
SIG_WIPE
BSF STROBE
RETURN
FIN_WIPP
BSF STROBE
BSF MODOHOLDW
CLRF EFECT_AUX
RETURN
FIN_WIPE
CALL LIMPIA_PANEL
clrf cursor
BCF STATUS,C ;BORRA EL BIT DE CARRY
CALL ENVIA_FRASE
INCF EFECT_AUX,1
MOVLW TIME_WIPE
XORWF EFECT_AUX,W
BTFSK STATUS,Z
GOTO SIG_HOLD_WIP
CLRF EFECT_AUX
```

82
35

```
    DECFSZ C_RETAEFEC,1    ;DECREMENTE Y SALTE SI CERO. DECREMENTA ERTA_HOLD Y SI
CERO, FIN HOLD
    GOTO SIG_HOLD_WIP
    GOTO FIN_HOLD_WIP    ;FIN DE CICLO HOLD
SIG_HOLD_WIP
    NOP
    NOP
    NOP
    BSF STROBE
    RETURN
FIN_HOLD_WIP
    MOVLW .7
    MOVWF ROTA_WIPE_UP
    MOVLW .1
    MOVWF COL_WIPE    ;GUARDA LA VARIABLE DE ROTACIÓN DE LAS COLUMNAS
    GOTO FIN_EFECTS
;*****
;*****
FIN_EFECTS
    CLRF BANDEMODO    ;LIMPIA MODOS DE EFECTOS
    CLRF EFECT_AUX
    bcf retaefec
    BCF FUENTE_FE
    BTFSC FUENTE
    BSF FUENTE_FE
BUSCA_CARFIN
    CLRWD
    CALL LEE_RAM
    BSF PCLATH,3
    CALL BUS_CAR_ESP
    clrf pclath
    CALL INC_DIRLH
    BCF STATUS,Z    ;PARA BUSCAR EL FINAL DE CARACTER. ULTIMA MODIFICACION.
    MOVLW 0xFE
    XORWF CHARACTER,W
    BTFSS STATUS,Z
    GOTO BUSCA_CARFIN
CARFIN
    BCF STATUS,Z
; CALL INC_DIRLH
    MOVF DIRL,W    ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
    MOVWF DIRINIL
    MOVF DIRH,W
    MOVWF DIRINI
    BSF STROBE
    BSF OE
    CALL BUSCADIRINI_ROTA
    RETURN
;*****
;*****
EF_ROTATE
    clrwdt
    clrf tiempo_desp
    CALL INC_DIRLH
    CLRF CURSOR
    clrf cursor_ini
; BTFSC MODOROTATE
; GOTO CICLECT
    bcf fuente_ini
    btfsc fuente_fe
    bsf fuente_ini
    MOVF DIRL,W    ;COLOCA EL PUNTERO DE LA FRASE EN EL INICIO DE LA FRASE CON EL
EFECTO
    MOVWF DIRINIL
    MOVF DIRH,W
    MOVWF DIRINI
    BSF MODOROTATE2
    CALL BUSCADIRINI_ROTA
```

33
32

```

RETURN
;***** SALIREF clrwdt ;Se
completó el total de columnas!
    bsf STROBE ;VISUALIZA LO QUE SE ENVIÓ
    bsf OE
    return
;*****
;*****
ENVIA_FRASE
    bcf fuente
    btfsc fuente_fe
    bsf fuente
    CLRF COLMNASL ;CONTADORES DE COLMNAS ENVIADAS
    CLRF COLMNASH
    MOVF NCTL,W
    MOVWF COLMNASLL ;CONTADORES DE COLMNAS ENVIADAS
    MOVF NCTH,W
    MOVWF COLMNASHH ;CONTADORES DE COLMNAS ENVIADAS
    CALL INC_DIRLH ;PARA INICIAR EN EL SIGUIENTE CARACTER AL DE IDENTIFICADOR DE
EFFECTO HOLD;
DIGITO
;    BCF STATUS,C ;SE ENVIA UN ESPACIO
;    CLRF DATOCOL
;    CALL ENVIA7
    CALL LEE_RAM ;VARIABLE CARACTER = CARACTER GUARDADO EN RAM A SER ENVIADO
;DIRECCIONADO POR DIRL Y DIRH
    movlw code_tnorm
    xorwf caracter,W
    btfsc STATUS,Z
    goto FRA_TEXTONORM ;Código de texto NORMAL
    movlw code_tneg
    xorwf caracter,W
    btfsc STATUS,Z
    goto FRA_TEXTONEGR ;Código de texto NEGRITA
    movlw code_tneg2
    xorwf caracter,W
    btfsc STATUS,Z
    goto FRA_TEXTONEGR2 ;Código de texto NEGRITA
    movf dirh,w
    movwf dirh_ht
    movf dirl,w
    movwf dirl_ht
    movlw code_hora1
    xorwf caracter,W
    btfsc STATUS,Z
    goto fra_mosthora1 ;Código de hora1
    movlw code_hora2
    xorwf caracter,W
    btfsc STATUS,Z
    goto FRA_MOSTHORA2 ;Código para mostrar la HORA
    movlw temperatura
    xorwf caracter,W
    btfsc STATUS,Z
    goto fra_mosttempe ;Código de temperatura
    call busca_fechacode
    btfsc fechacode
    goto SIG_DIGITO_EF ;Código para mostrar la FECHA
    MOVLW CODE_FINEFECTO
    xorwf caracter,W
    btfsc STATUS,Z
    goto FN_DI ;Código para FIN de mensajes!
    movlw code_finmensaje
    xorwf caracter,W
    btfsc STATUS,Z
    goto FN_DI ;Código para FIN de mensajes!
    call code_velo ;codigo de velocidad
    btfss codevelo
    goto conti_digi ;DIGITO ASCII. VISUALIZA EN EL EFECTO CORRESP

```

```

    goto    set_code_velo
;   GOTO    DIGITO
SIG_DIGITO_EF
    CALL    INC_DIRLH
    BTFSC   TEMPE_HR_FE
    CALL    ACTUALIZA_H_T ;ACTUALIZA
    clrf    cursor
    GOTO    DIGITO
;*****
set_code_velo
    btfsc   retaefec
    goto    fn_setvel
    bsf     retaefec
    btfsc   modohold
    goto    set_velhold
    btfsc   modoflash
    goto    set_velflash
    btfsc   modoscroll
    goto    set_velscroll
    btfsc   modowipe_up
    goto    set_velwipe
;   btfsc   modowide
;   goto    set_velwide
    goto    fn_setvel
set_velhold
    movf    codascii,w
    MOVWF   RETA_HOLD
    MOVWF   C_RETA_EFEC
    goto    fn_setvel
set_velflash
    movf    codascii,w
    MOVWF   RETA_FLASH
    BCF     STATUS,C
    RLF     RETA_FLASH,W ;MULTIPLICACION POR 2
    MOVWF   RETA_FLASH
    MOVWF   C_RETA_EFEC
    goto    fn_setvel
set_velscroll
    movf    codascii,w
    MOVWF   RET_SCRL
    BCF     STATUS,C
    RLF     RET_SCRL,W
    MOVWF   RET_SCRL
    MOVWF   C_RETA_EFEC
    goto    fn_setvel
set_velwipe
    movf    codascii,w
    MOVWF   RET_WIPE
    BCF     STATUS,C
    RLF     RET_WIPE,W
    ADDWF   RET_WIPE,W ;*3
    MOVWF   RET_WIPE
    MOVWF   C_RETA_EFEC
    goto    fn_setvel
;set_velwide
;   movf    codascii,w
;   MOVWF   RET_WIDE
;   BCF     STATUS,C
;   RLF     RET_WIDE,W
;   ADDWF   RET_WIDE,W ;*3
;   MOVWF   RET_WIDE
;   MOVWF   C_RETA_EFEC
;   goto    fn_setvel
fn_setvel
    CALL    INC_DIRLH
    CALL    INC_DIRLH ;para eliminar el espacio que llega con el código de velocidad
    goto    digito
;*****
;

```

35
28

```

set_codevelo_rot
; MOVF CODASCII,W
; XORWF TPO_DESP,1
; BTFSS STATUS,Z
; CLRF TIEMPO_DESP
; MOVF CODASCII,W
; MOVWF TPO_DESP
btfss time_desp
return
MOVF CODASCII,W
MOVWF TPO_DESP
CLRF TIEMPO_DESP
return
;+++++
FRA_TEXTONORM
call code_tnorm_cod ;
goto SIG_DIGITO_EF
;*****
FRA_TEXTONEGR
call code_tneg_cod
goto SIG_DIGITO_EF
;*****
FRA_TEXTONEGR2
call code_tneg2_cod
goto SIG_DIGITO_EF
;*****
FRA_MOSTHORA1
call CODE_HORA1_COD
goto SIG_DIGITO_EF ;No se ha terminado el refresco, entonces continúa leyendo datos de RAM!
;*****
FRA_MOSTHORA2
CALL CODE_HORA2_COD
goto SIG_DIGITO_EF ;No se ha terminado el refresco, entonces continúa leyendo datos de RAM!
;*****
fra_mosttempe
CALL temperatura_cod
goto SIG_DIGITO_EF
;*****
CONTI_DIGI
BCF STATUS,Z
CALL INC_DIRLH
CALL CALCULA_OFFSET ;CALCULA POSICIÓN OFFSET EN LA TABLA DE CARACTERES DEL
FINAL, PARA EL CARACTER 'CARACTER'
movf DespOffsetL,W
movwf DESPLAZL
movf DespOffsetH,W
movwf DESPLAZH
movf DespOffsetL,W
addwf cursor,W
movwf DESPLAZL
btfsc STATUS,C
incf DESPLAZH,1
bcf fin_digit
COL ;CICLO DE BUSQUEDA DE LAS COLUMNAS DEL CARACTER
btfsc fin_digit
GOTO DIGITO
clrwdt
INCF COLMNASL,1 ;CONTADOR DE INCREMENTO DE COLUMNAS
BTFSC STATUS,Z
INCF COLMNASH,1
BCF STATUS,Z
MOVLW .0
XORWF COLMNASLL,w
BTFSC STATUS,Z
DECF COLMNASHH,1
DECF COLMNASLL,1 ;CONTADOR DE COLUMNAS SOBRANTES
MOVLW 0xFF ;SE ACABARON LAS COLUMNAS?
XORWF COLMNASLL,W

```

36
35

```
BTFSS STATUS,Z
GOTO SIGA_COLL
BCF STATUS,Z
MOVLW 0xFF
XORWF COLMNASHH,W
BTFSC STATUS,Z
RETURN ;SI YA NO HAY COLUMNAS DISPONIBLES RETORNAR
; GOTO SIGA_COLL
SIGA_COLL
    bcf fin_digit
    call BUSCACOL ;
    BTFSC DATOCOL,0
    bsf fin_digit ;indicador de fin de digito
    btfsc fin_digit
    clrf datocol
; btfsc modowide
; goto env_wide
BTFSC MODOSCROLL ;VERIFICAR SI ESTA ACTIVADO EL SCROLL
GOTO ENV_SCROLL
BTFSC MODOWIPE_UP ;VERIFICAR SI ESTA ACTIVADO EL
GOTO ENV_WIPE
CALL ENVIA7 ;MODO FLASH O HOLD ACTIVADO
INCF DESPLAZL,1
BTFSC STATUS,Z
INCF DESPLAZH,1
BCF STATUS,Z
GOTO COL
ENV_SCROLL ;MODO SCROLL
;REALIZAR CORRIMIENTO CON COL_AUX(PARA GUARDAR EL RESULTADO DE LA ROTACION) Y
COL_SCROLL(NUMERO DE ROTACIONES)
    MOVF COL_SCROLL,W
    MOVWF CARAUX2
    CLRF COL_AUX
ROT_SCRO
    BCF STATUS,C
    RRF DATOCOL,1
    RRF COL_AUX,1
    DECFSZ CARAUX2,1
    GOTO ROT_SCRO
    MOVF COL_AUX,W
    MOVWF DATOCOL
    CALL ENVIA7
    INCF DESPLAZL,1
    BTFSC STATUS,Z
    INCF DESPLAZH,1
    GOTO COL
ENV_WIPE
    MOVF COL_WIPE,W
    MOVWF CARAUX2
    CLRF COL_AUX
ROT_WIPE
    BSF STATUS,C
    RRF COL_AUX,1
    DECFSZ CARAUX2,1
    GOTO ROT_WIPE
    MOVF COL_AUX,W
    ANDWF DATOCOL,W
    MOVWF DATOCOL
    CALL ENVIA7
    INCF DESPLAZL,1
    BTFSC STATUS,Z
    INCF DESPLAZH,1
    GOTO COL
;*****
;ENV_WIDE
; CALL ENVIA7
; INCF DESPLAZL,1
; BTFSC STATUS,Z
```

37
36

```
; INCF DESPLAZH,1
; GOTO COL
;*****
;
FN_DI
    MOVWF COLMNASL      ;PARA EL MODO FLASH,
    BCF STATUS,C ;DETERMINA EL NUMERO DE COLUMNAS DE ESPACIO
    RRF COLMNASHH,1      ;DIVISIÓN POR 2
    RRF COLMNASLL,1      ;DIVISIÓN POR 2
    BCF STATUS,Z
    MOVF COLMNASLL,W
    IORWF COLMNASHH,W
    XORLW 0x00
    BTFSC STATUS,Z
    RETURN
;centrado
    CALL ENV_ESPACIOS ;PRUEBABA
    MOVLW TIME_REFRESH
    CALL RETA
    RETURN
;*****
;
ENV_ESPACIOS
    BCF STATUS,C ;BORRA EL BIT DE CARRY
    CLRF DATOCOL
    ESPA      ;ESPACIOS
    CALL ENVIA7
    DECFSZ COLMNASLL,1
    GOTO ESPA
    BCF STATUS,Z
    MOVF COLMNASLL,W
    IORWF COLMNASHH,W
    BTFSC STATUS,Z ;SI YA NO HAY COLUMNAS DISPONIBLES RETORNAR
    RETURN
    DECF COLMNASHH,1
    GOTO ESPA
;*****
;
LIMPIA_PANEL      ;ENVIA COLUMNAS EN LANCO A TODO EL PANEL
    MOVF NCTL,W
    MOVWF NCOLUMNAS_L
    MOVF NCTH,W
    MOVWF NCOLUMNAS_H
    CLRF DATOCOL
    BCF STATUS,C ;BORRA EL BIT DE CARRY
LIM_PA
    CALL ENVIA7
    DECFSZ NCOLUMNAS_L,1
    GOTO LIM_PA
    BCF STATUS,Z
    MOVLW .0
    XORWF NCOLUMNAS_H,W
    BTFSC STATUS,Z ;SALTE SI DIFERENTES
    RETURN
    DECF NCOLUMNAS_H,1
    MOVLW 0xFF
    MOVWF NCOLUMNAS_L
    GOTO LIM_PA
;*****
;
ENVCOLCAR
    clrwdt
    bcf aborto
    call calcula_offset ;CALCULA POSICIÓN OFFSET EN LA TABLA DE CARACTERES DEL
FINAL,PARA EL CARACTER 'CARACTER'
    movf DespOffsetL,W
    movwf DESPLAZL
    movf DespOffsetH,W
    movwf DESPLAZH
    movf DespOffsetL,W
    addwf cursor,W
    movwf DESPLAZL
```

```

btfsc STATUS,C
incf DESPLAZH,1
SigaEnvCol
clrwdt
call BUSCACOL
btfsc DATOCOL,0 ;Es fin de caracter?
bsf fin_car ;Sí, entonces se activa bandera de fin de caracter.
call ENVIA7 ;Envía columna apuntada por cursor del carácter ASCII correspondiente
btfss fin_car ;Está habilitado fin de caracter?
goto INRCOLX ;No
bcf fin_car
incf contcolenvL,1 ;Se incrementa contador de columnas enviadas.
btfsc STATUS,Z
incf contcolenvH,1
; [Cont column enviadas] = [Total de columnas del panel para llenarlo]?
movf NCTH,W
xorwf contcolenvH,W
btfss STATUS,Z
goto SalEnvCar
bcf status,z
movf NCTL,W
xorwf contcolenvL,W
btfss STATUS,Z
goto SalEnvCar
bsf aborto ;Se completó el total de columnas!
goto SalEnvCar
INRCOLX
clrwdt
incf cursor,1
movf DespOffsetL,W
addwf cursor,W
movwf DESPLAZL
btfsc STATUS,C
incf DESPLAZH,1
incf contcolenvL,1
btfsc STATUS,Z
incf contcolenvH,1
movf NCTH,W ;[Cont column enviadas] = [Total de columnas del panel para llenarlo]?
xorwf contcolenvH,W
btfss STATUS,Z
goto SigaEnvCol
movf NCTL,W
xorwf contcolenvL,W
btfss STATUS,Z
goto SigaEnvCol
bsf aborto ;Se completó el total de columnas!
SalEnvCar
clrwdt
return
;*****
;CALCULA POSICIÓN OFFSET EN LA TABLA:
calcula_offset
clrwdt ;CARACTER=CARACTER
clrf DespOffsetH
movlw ''
subwf caracter,W ;[código ascii del caracter - 32 (ascii de ESPACIO)]
movwf DespOffsetL
bcf STATUS,C
rlf DespOffsetL,1
rlf DespOffsetH,1
bcf STATUS,C
rlf DespOffsetL,1
rlf DespOffsetH,1
bcf STATUS,C
rlf DespOffsetL,1
rlf DespOffsetH,1 ;[código ascii del caracter - 32 ] * 8
movlw .32 ;{[código ascii del caracter - 32 ] * 8 } + 32
addwf DespOffsetL,1

```

39
38

```
btfsc STATUS,C
incf DespOffsetH,1
return
;*****
;
BUSCACOL
clrwdt
btfsc fuente ;Determina tipo de fuente: 0 >> Normal, 1 >> Negrilla.
goto BUSCANEG
BUSCANORM
clrwdt
movlw LOW CAR_NORM
addwf DESPLAZL,W
movwf caraux
movlw HIGH CAR_NORM
addwf DESPLAZH,W
movwf PCLATHtemp
movf PCLATHtemp,W
bsf PCLATH,4
call CAR_NORM ;Se encuentra en página 1 de memoria de programa.
clrf PCLATH
goto SIG555
BUSCANEG
clrwdt
movlw LOW CAR_NEG
addwf DESPLAZL,W
movwf caraux
movlw HIGH CAR_NEG
addwf DESPLAZH,W
movwf PCLATHtemp
movf PCLATHtemp,W
bsf PCLATH,4
bsf PCLATH,2
call CAR_NEG ;Se encuentra en página 2 de memoria de programa.
clrf PCLATH
SIG555 movwf DATOCOL
return
;*****
;*****
;
BUSCADIRINI
clrf DirIniH
clrf DirIniL
clrf cursor_ini
BCF FUENTE
BCF FUENTE_INI
BTFSC MODOROTATE
GOTO SEEKI
BUSCADIRINI_ROTA
bcf colvacias ;Sí existen columnas vacías.
bcf STATUS,C
movlw .1
subwf NCTL,W
movwf NCVL
btfss STATUS,Z
goto MIRANEGX2
movlw .0
xorwf NCTH,W
btfss STATUS,Z
goto SALRESTA12
clrf NCVH
bsf colvacias ;No hay columnas vacías!!
goto SEEKDIRINI
MIRANEGX2
btfsc STATUS,C ;resultado de resta anterior negativo?
goto SALRESTA12 ;No
decf NCTH,W ;Sí
movwf NCVH
goto SEEKDIRINI
SALRESTA12
```

```

movf NCTH,W
movwf NCVH
SEEKI
    clrf bandemod
; bsf modorotate
    bsf modorotate2
SEEKDIRINI
    clrwdt
    movf DirIniL,W
    movwf DIRL
    movwf DIRL_HT
    movf DirIniH,W
    movwf DIRH
    movwf DIRH_HT
    call LEE_RAM
    movlw code_tnorm ;Código de texto NORMAL
    xorwf character,W
    btfss STATUS,Z
    goto salx111
    call code_tnorm_cod
    goto SIGABUSDIRINI
salx111
    clrwdt
    movlw code_tneg ;Código de texto NEGRITA
    xorwf character,W
    btfss STATUS,Z
    goto salx211
    call code_tneg_cod
    goto SIGABUSDIRINI
salx211
    clrwdt
    movlw code_tneg2 ;Código de texto NEGRITA2
    xorwf character,W
    btfss STATUS,Z
    goto salx221
    call code_tneg2_cod
    goto SIGABUSDIRINI
salx221
    clrwdt
    movlw code_hora1 ;Código para mostrar la HORA
    xorwf character,W
    btfss STATUS,Z
    goto salx411
    call code_hora1_cod
    goto SIGABUSDIRINI
salx411
    clrwdt
    movlw code_hora2 ;Código para mostrar la HORA2
    xorwf character,W
    btfss STATUS,Z
    goto salx441
    call code_hora2_cod
    goto SIGABUSDIRINI
salx441
    CALL BUSCA_FECHACODE
    BTFSC FECHACODE
    goto SIGABUSDIRINI
    movlw temperatura ;Código para mostrar la HORA2
    xorwf character,W
    btfss STATUS,Z
    goto salx411
    call temperatura_cod
    goto SIGABUSDIRINI
salx411
    movlw car_rotate
    xorwf character,W
    btfss STATUS,Z
    goto salx511

```

41
60

```
call car_rotate_cod
goto SIGABUSDIRINI
salxx511
call code_velo      ;codigo de velocidad
btfss codevelo
goto salxx5111
CALL set_codevelo_rot
goto SIGABUSDIRINI
salxx5111
movlw code_finefecto      ;Código para FIN DE CHARACTER
xorwf character,W
btfsc STATUS,Z
goto SIGABUSDIRINI
movlw code_finmensaje      ;Código para FIN de mensajes!
xorwf character,W
btfss STATUS,Z
goto SALECCC      ;Entonces es caracter ascii ! ;[No se chequea si es código válido, puesto...
                    ;que ya está garantizado que al menos es código de ESPACIO !]
bsf    pclath,3
bsf    pclath,2
call SETUPRAM2      ;No se encontró código de caracter ascii, entonces graba mensaje por defecto.
CLRF   PCLATH
goto   SALECCC      ;Entonces es caracter ascii ! ;[No se chequea si es código válido, puesto...
                    ;que ya está garantizado que al menos es código de ESPACIO !]
SIGABUSDIRINI
incf   DirIniL,1
btfsc  STATUS,Z
incf   DirIniH,1
goto   SEEKDIRINI
SALECCC return
,*****
CODE_TNORM_COD      ;Código de texto NORMAL
bcf     fuente
bcf     fuente_ini
bcf     fuente_ini2
RETURN
CODE_TNEG_COD       ;Código de texto NEGRITA
bsf     fuente
bcf     fuente_ini2
bsf     fuente_ini
RETURN
CODE_TNEG2_COD      ;Código de texto NEGRITA2
bsf     fuente2
bcf     fuente_ini
bsf     fuente_ini2
RETURN
CODE_HORA1_COD      ;Código de HORA1
movf    DIRH_HT,W
movwf   DIRVERHR_H
movf    Dirl_HT,W
movwf   DIRVERHR_L
incf    DIRVERHR_L,1
btfsc   STATUS,Z
incf    DIRVERHR_H,1
BCF     HORA_MILITAR
BSF     TEMPE_HR_FE
RETURN
CODE_HORA2_COD      ;Código para mostrar la HORA2
movf    DIRH_HT,W
movwf   DIRVERHR_H
movf    Dirl_HT,W
movwf   DIRVERHR_L
incf    DIRVERHR_L,1
btfsc   STATUS,Z
incf    DIRVERHR_H,1
BSF     HORA_MILITAR
BSF     TEMPE_HR_FE
RETURN
```

BUSCA_FECHACODE
 BCF FECHACODE
 movlw code_fecha1 ;Código para mostrar la FECHA01
 xorwf character,W
 btfss STATUS,Z
 goto salx5110
 BCF FECHA02
 BSF FECHA01
 GOTO SALX5112

salx5110
 movlw code_fecha2 ;Código para mostrar la FECHA02
 xorwf character,W
 btfss STATUS,Z
 goto salx5111
 BCF FECHA01
 BSF FECHA02
 GOTO SALX5112

salx5111
 movlw code_fecha3 ;Código para mostrar la FECHA03
 xorwf character,W
 btfss STATUS,Z
 goto salx511
 BCF FECHA01
 BCF FECHA02

salx5112
 movf DIRH_HT,W
 movwf DIRVERFE_H
 movf DIRM_HT,W
 movwf DIRVERFE_L
 incf DIRVERFE_L,1
 btfsc STATUS,Z
 incf DIRVERFE_H,1
 BSF FECHACODE
 BSF TEMPE_HR_FE

SALX511
 RETURN

TEMPERATURA_COD
 movf DIRH_HT,W
 bsf STATUS,RP0
 movwf DIRVERTMP_H_b2
 bcf STATUS,RP0
 movf DIRM_HT,W
 bsf STATUS,RP0
 movwf DIRVERTMP_L_b2
 incf DIRVERTMP_L_b2,1
 btfsc STATUS,Z
 incf DIRVERTMP_H_b2,1
 bcf STATUS,RP0
 BSF TEMPE_HR_FE
 RETURN

CAR_ROTATE_COD
 clrf bandemod
 CLRF CURSOR
 clrf cursor_ini
 bsf modorotate2 ;activa modo rotate
 ; CALL BUSCADIRINI_ROTA
 RETURN

*****;rutina para buscar codigo de

velocidad
 code_velo
 BCF codevelo
 movlw code_velmin
 movwf codascii
 codevel2
 CLRWDI
 movf codascii,W
 xorwf character,W
 btfsc STATUS,Z

43
42

```

goto si_codevel
movlw code_velmax
xorwf codascii,W
btfsc STATUS,Z
goto fin_codevelo
incf codascii,1
goto codevel2
si_codevel
    bsf codevelo
; incf codascii,1
    movlw code_velmin
    subwf codascii,w
    sublw .10
    movwf codascii
; CALL INC_DIRLH ;
; movlw 0x5B ;ESPACIO CHIQUITO QUE SOBREESCRIBE EL QUE ENVIA EL SOFTWARE
; movwf DATOX
; call ESC_RAM
; decf DIRL,1
; btfsc STATUS,Z
; incf DIRH,1
fin_codevelo
    return
;*****
;SETUPRAM
; clrwdt
; bcf colvacias ;Sí hay columnas vacías.
; clrf DIRH
; movlw .1
; movwf DIRL
; movlw '!'
; movwf DATOX
; call ESC_RAM
; CALL INC_DIRLH
; movlw 0xFC
; movwf DATOX
; call ESC_RAM ; RAM 0001H = !
; ;Establece número de columnas vacías [NCT - 2]: 0002H = FF
; bcf STATUS,C
; movlw .5
; subwf NCTL,W
; movwf NCVL
; btfss STATUS,Z
; goto MIRANEGXX
; movlw .0
; xorwf NCTH,W
; btfss STATUS,Z
; goto SALRESTA22
; clrf NCVH
; bsf colvacias ;No hay columnas vacías!!
; goto ESTATICO
;MIRANEGXX
; btfsc STATUS,C ;resultado de resta anterior negativo?
; goto SALRESTA22 ;No
; decf NCTH,W ;Sí
; movwf NCVH
; goto ESTATICO
;SALRESTA22
; movf NCTH,W
; movwf NCVH
;ESTATICO
; bcf modoRotate
; clrf DirIniH
; clrf DirIniL
; clrf cursor_ini
; return
;*****
;*****
;

```

ACTUALIZA_H_T

```

movf DIRH,W
bsf STATUS,RP0
movwf DIRH_TMP_b2
bcf STATUS,RP0
movf DIRL,W

bsf STATUS,RP0
movwf DIRL_TMP_b2
bcf STATUS,RP0
bsf pclath,3
bsf pclath,2
call LEEDS1307
call ACTUALIZHR
call ACTUALIZFE
bsf pclath,0
bsf pclath,3
CALL ACTUALIZTMP
CLRF PCLATH
bsf STATUS,RP0
movf DIRH_TMP_b2,W
bcf STATUS,RP0
movwf DIRH
bsf STATUS,RP0
movf DIRL_TMP_b2,W
bcf STATUS,RP0
movwf DIRL
BCF TEMPE_HR_FE ;DESACTIVE BANDERA DE ACTUALIZACION
clrwdt
RETURN
nop
nop
nop

```

ACTUALIZHR

```

clrwdt
movf DIRVERHR_H,W
xorlw 0x00
btfss STATUS,Z
goto GRAHR
movf DIRVERHR_L,W
xorlw 0x00
btfss STATUS,Z
goto GRAHR
goto SALACTHR ;No existe dir RAM para insertar la hora
GRAHR
movf DIRVERHR_H,W
movwf DIRH
movf DIRVERHR_L,W
movwf DIRL
BCF HORA_AMPM
BTFSC HORA,5
BSF HORA_AMPM ;HORA PM
BTFSC HORA_MILITAR
GOTO HORA_MILI
HORA_NORMAL
btfsc hora,4
goto DEC
movlw 0x5B
goto SIGXXX
DEC movlw '1'
SIGXXX
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
movf hora,W
andlw 0x0F
addlw 0x30

```

45
40x

```
movwf DATOX
call ESC_RAM
GOTO SIGUE_MINS
HORA_MILI
; BCF HORA_MILITAR
BCF STATUS,Z
MOVF HORA,W
ANDLW 0x0F
XORLW 0x02
BTFSS STATUS,Z
GOTO SIGXXX1
BTFSS HORA,4
GOTO SIGXXX1
BTFSC HORA_AMPM
GOTO HORA_NORMAL
GOTO SIGXXX11
SIGXXX1
BTFSS HORA_AMPM
GOTO HORA_NORMAL
SIGXXX11
BCF HORA,7
INCF HORA,1
BCF STATUS,Z
MOVF HORA,W
ANDLW 0x0F
XORLW 0x0A
BTFSC STATUS,Z
CALL PRINT2
INCF HORA,1
BCF STATUS,Z
MOVF HORA,W
ANDLW 0x0F
XORLW 0x0A
BTFSS STATUS,Z
GOTO SIGXXX2
CALL PRINT2
GOTO SIGXXX3
PRINT2
movlw '2'
movwf DATOX
call ESC_RAM
MOVLW 0xF0
ANDWF HORA,1
BSF HORA,7
RETURN
SIGXXX2
btfsc hora,7
GOTO SIGXXX3
btfss hora,4
GOTO PRINT1
PRINT3
BCF STATUS,Z
MOVF HORA,W
ANDLW 0x0F
XORLW 0x04
BTFSC STATUS,Z
goto print00
movlw '2'
movwf DATOX
call ESC_RAM
GOTO SIGXXX3
PRINT1
movlw '1'
movwf DATOX
call ESC_RAM
goto SIGXXX3
print00
movlw '0'
```

AC
45

```
movwf DATOX
call ESC_RAM
call inc_dirlh
movlw '0'
movwf DATOX
call ESC_RAM
goto sigue_mins
SIGXXX3
CALL INC_DIRLH
movf hora,W
andlw 0x0F
addlw 0x30
movwf DATOX
call ESC_RAM
SIGUE_MINS
CALL INC_DIRLH
movlw '.'
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
swapf minutos,W
andlw 0x0F
addlw 0x30
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
movf minutos,W
andlw 0x0F
addlw 0x30
movwf DATOX
call ESC_RAM
clrwdt
SALACTHR
clrwdt
return
,*****
,*****
,
ACTUALIZFE
clrwdt
movf DIRVERFE_H,W
xorlw 0x00
btfss STATUS,Z
goto GRAFE
movf DIRVERFE_L,W
xorlw 0x00
btfsc STATUS,Z
RETURN ;No existe dir RAM para insertar la fecha
GRAFE
movf DIRVERFE_H,W
movwf DIRH
movf DIRVERFE_L,W
movwf DIRL
BTFSC FECHA01
GOTO FECHA_01
BTFSC FECHA02
GOTO FECHA_02
GOTO FECHA_03 ;POR DEFECTO
FECHA_01
CALL ACT_DIA
CALL ACT_DE
CALL ACT_MES_STR
CALL ACT_DE
CALL ACT_AÑO
; goto fin_fecha
RETURN
FECHA_02
CALL ACT_MES_STR
CALL ACT_ESPA
```

```

CALL ACT_DIA
CALL ACT_DE
CALL ACT_AÑO
; goto fin_fecha
RETURN
FECHA_03
CALL ACT_ESPA
CALL ACT_DIA
CALL ACT_SLASH
CALL ACT_MES_NUM
CALL ACT_SLASH
CALL ACT_AÑO2
;goto fin_fecha
RETURN
fin_fecha
return
,*****
,
ACT_MES_NUM
SWAPF MES,W
ANDLW 0x01
ADDLW 0x30
MOVWF DATOX
CALL ESC_RAM
CALL INC_DIRLH
MOVF MES,W
ANDLW 0x0F
ADDLW 0x30
MOVWF DATOX
CALL ESC_RAM
CALL INC_DIRLH
RETURN
,*****
,
ACT_MES_STR
movlw 0x01
xorwf mes,W
btfsc STATUS,Z
goto ENERO
movlw 0x02
xorwf mes,W
btfsc STATUS,Z
goto FEBRERO
movlw 0x03
xorwf mes,W
btfsc STATUS,Z
goto MARZO
movlw 0x04
xorwf mes,W
btfsc STATUS,Z
goto ABRIL
movlw 0x05
xorwf mes,W
btfsc STATUS,Z
goto MAYO
movlw 0x06
xorwf mes,W
btfsc STATUS,Z
goto JUNIO
movlw 0x07
xorwf mes,W
btfsc STATUS,Z
goto JULIO
movlw 0x08
xorwf mes,W
btfsc STATUS,Z
goto AGOSTO
movlw 0x09
xorwf mes,W
btfsc STATUS,Z

```

```

goto SEPTIEMBRE
movlw 0x10
xorwf mes,W
btfsc STATUS,Z
goto OCTUBRE
movlw 0x11
xorwf mes,W
btfsc STATUS,Z
goto NOVIEMBRE
movlw 0x12
xorwf mes,W
btfsc STATUS,Z
goto DICIEMBRE
ENERO movlw .0
goto ESCRIBEMES
FEBRERO movlw .4
goto ESCRIBEMES
MARZO movlw .8
goto ESCRIBEMES
ABRIL movlw .12
goto ESCRIBEMES
MAYO movlw .16
goto ESCRIBEMES
JUNIO movlw .20
goto ESCRIBEMES
JULIO movlw .24
goto ESCRIBEMES
AGOSTO movlw .28
goto ESCRIBEMES
SEPTIEMBRE
movlw .32
goto ESCRIBEMES
OCTUBRE movlw .36
goto ESCRIBEMES
NOVIEMBRE
movlw .40
goto ESCRIBEMES
DICIEMBRE
movlw .44
ESCRIBEMES
movwf posic_car
movlw .3
movwf CUATRO
CICGRABMES
movf posic_car,W
call CadenaMes
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
incf posic_car,1
decfsz CUATRO,1
goto CICGRABMES
RETURN
*****
,
ACT_DIA ;Determina día del mes:
swapf dia_mes,W
andlw 0x03 ;Captura decenas de DIA_MES
addlw 0x30
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
movf dia_mes,W
andlw 0x0F ;Captura unidades de DIA_MES
addlw 0x30
movwf DATOX
call ESC_RAM
CALL INC_DIRLH
RETURN

```

```

,*****
;
ACT_de
    CALL    ACT_ESPA
    movlw  'd'
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    movlw  'e'
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    CALL    ACT_ESPA
    RETURN
,*****
;
ACT_ESPA
    movlw  0x7B
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    RETURN
,*****
;
ACT_SLASH
    movlw  '/'
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    RETURN
,*****
;
;Determina Año:
ACT_AÑO
    movlw  '2'
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    movlw  '0'
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
ACT_AÑO2
    swapf  an_o,W
    andlw  0x0F    ;Captura decenas de AÑO
    addlw  0x30
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    movf  an_o,W
    andlw  0x0F    ;Captura unidades de AÑO
    addlw  0x30
    movwf  DATOX
    call  ESC_RAM
    CALL    INC_DIRLH
    clrwdt
    return
,*****
;
,*****
;
,*****
;
INC_DIRLH
    incf  DIRL,1
    btfsc STATUS,Z
    incf  DIRH,1
    RETURN
,*****
;
RETA
    MOVWF RETARD
RERE
    CLRWDT
; NOP
; NOP

```

80
49

```
; NOP
; DECFSZ RETARD,1
; GOTO RERE
; RETURN
;*****
;Envía un dato de 8 bits (Direccion ALTA para RAM) al registro de desplazamiento 4094.
;ENVDIRH1 movwf DIR_RAM_H
; movlw .8
; movwf contbits
;TX4094b2 bcf CLK94
; clrwdt
; bcf DAT94 ;Coloca en cero línea de datos del 4094
; btfsc DIR_RAM_H,7
; bsf DAT94 ;Coloca en alto línea de datos del 4094
; nop
; clrwdt
; bsf CLK94 ;Pulso de reloj para los registros 4094
; rlf DIR_RAM_H,1 ;Rota byte que se está enviando (DATOCOL)
; decfsz contbits,1
; goto TX4094b2
; return
;*****
;Envía un dato de 8 bits (Direccion ALTA para RAM) al registro de desplazamiento 4094.
ENVDIRH
;-----
;Envía un dato de 7 bits a los registros de desplazamiento 4094.
ENVIA7
; BSF STATUS,RP0
; movlw B'11111000' ;X X X RTC X STRB CLK DATA (PANEL)
; movwf TRISA
; BCF STATUS,RP0
; movlw .7
; movwf contbits
TX4094
; bcf CLOCK
; clrwdt
; bcf DATAS ;Coloca en cero línea de datos del 4094
; btfsc DATOCOL,7
; bsf DATAS ;Coloca en alto línea de datos del 4094
; nop
; clrwdt
; bsf CLOCK ;Pulso de reloj para los registros 4094
; rlf DATOCOL,1 ;Rota byte que se está enviando (DATOCOL)
; decfsz contbits,1
; goto TX4094
; BCF CLOCK
; CLRF DATOCOL
; return
;*****
lee_raml
; movlw LOW FRASE
; addwf DIRL,W
; movwf caraux
; movlw HIGH FRASE
; addwf DIRH,W
; movwf PCLATHtemp
; movf PCLATHtemp,W
; bsf PCLATH,3
; bsf PCLATH,4
; call FRASE ;Se encuentra en página 2 de memoria de programa.
; movwf CHARACTER
; clrf PCLATH
; return
LEE_RAM
; clrwdt
; bcf STR94
; call ENVDIRHL1
; bsf STR94
```

51
50

```

bsf    STATUS,RP0
movlw  0xFF
movwf  TRISB
bcf    STATUS,RP0
bsf    WE    ;Lectura de RAM
bcf    OER    ;Salida de datos de RAM habilitada
bcf    CS    ;Selecciona RAM
nop
nop
movf   PORTB,W
movwf  CHARACTER
nop
bsf    CS    ;RAM inhabilitada
bsf    OER    ;Entrada de datos a RAM deshabilitada
BCF    STR94 ;PARA BAJAR EL PULSO
return

```

```

;-----
;*****
;Envía un dato de 15 bits (Direccion ALTA para RAM) al registro de desplazamiento 4094.

```

```

ENVDIRHL1
    MOVF  DIRH,W
    movwf DIR_RAM_HL
    movlw .8
    movwf contbits
TX4094BH1
    bcf   CLK94
    clrwdt
    bcf   DAT94    ;Coloca en cero línea de datos del 4094
    btfsc DIR_RAM_HL,7
    bsf   DAT94    ;Coloca en alto línea de datos del 4094
    nop
    clrwdt
    bsf   CLK94    ;Pulso de reloj para los registros 4094
    rlf   DIR_RAM_HL,1 ;Rota byte que se está enviando (DATOCOL)
    decfsz contbits,1
    goto  TX4094bH1
    MOVF  DIRL,W
    movwf DIR_RAM_HL
    movlw .8
    movwf contbits

```

```

TX4094BL1
    bcf   CLK94
    clrwdt
    bcf   DAT94    ;Coloca en cero línea de datos del 4094
    btfsc DIR_RAM_HL,7
    bsf   DAT94    ;Coloca en alto línea de datos del 4094
    nop
    clrwdt
    bsf   CLK94    ;Pulso de reloj para los registros 4094
    rlf   DIR_RAM_HL,1 ;Rota byte que se está enviando (DATOCOL)
    decfsz contbits,1
    goto  TX4094bL1
    NOP
    NOP
    NOP
    BCF   CLK94
    BCF   DAT94
    return

```

```

;-----
Lee_EEP clrwdt
    bsf   STATUS,RP1
    bsf   STATUS,RP0 ;Banco 3 de RAM
    bcf   EECON1,EEPGD ;PROGRAM/DATA SELECT BIT; ACCESO A LA MEMORIA EEPROM DE
DATOS
    bsf   EECON1,RD    ;INICIA LECTURA DE LA MEMORIA EEPROM DE DATOS
    bcf   STATUS,RP0 ;Banco 2 de RAM
    movf  EEDATA,w
    bcf   STATUS,RP1 ;Banco 0 de RAM

```

```
movwf INDF
bsf STATUS,RP1
bcf STATUS,RP0 ;Banco 2 de RAM
incf EEADR,1
bcf STATUS,RP1 ;Banco 0 de RAM
incf FSR,1
decfsz cuantos,1
goto Lee_EEP
return
;*****
;*****
ESC_RAM clrwdt ;ESCRIBE EN RAM EXTERNA 'DATOX'. PARTE ALTA 'DIRH' ES
DIRECCIONADA DE FORMA SERIAL POR UN 4094
bcf STR94 ;EN LA RUTINA ENVIDIRH Y LA PARTE BAJA 'DIRL' ES DIRECCIONADA POR EL
PUERTO D
; movf DIRH,W
call ENVDIRHL1
bsf STR94
nop
nop
nop
nop
nop
; movf DIRL,W
; movwf PORTD
bsf STATUS,RP0
movlw 0x00
movwf TRISB
bcf STATUS,RP0
movf DATOX,W
movwf PORTB
bcf CS ;Selecciona RAM
bsf OER ;salida de datos a RAM deshabilitada
bcf WE ;escritura en RAM
nop
nop
CLRWDT
NOP
nop
nop
nop
nop
nop
nop
NOP
bsf CS ;RAM inhabilitada
bsf WE ;Lectura de RAM
; bsf OER ;Entrada de datos a RAM deshabilitada
NOP
nop
nop
nop
BCF STR94
return
;*****
;*****
;*****
;*****
;*****
;*****
;*****
nop
nop
nop
nop
nop
nop
nop
```

52

[illegible]

~~S4~~
S3

[illegible]

85
94

[illegible]

57
56

```
goto BUSX211
CLRF PCLATH
call code_tneg_cod
BSF PCLATH,3
goto BUSXx5111
BUSX211
    clrwdt
    movlw code_tneg2    ;Código de texto NEGRITA2
    xorwf character,W
    btfss STATUS,Z
    goto BUSX2211
    CLRF PCLATH
    call code_tneg2_cod
    BSF PCLATH,3
    goto BUSXx5111
BUSX2211
    clrwdt
    movlw code_hora1    ;Código para mostrar la HORA
    xorwf character,W
    btfss STATUS,Z
    goto BUSX411
    CLRF PCLATH
    call code_hora1_cod
    BSF PCLATH,3
    goto BUSXx5111
BUSX411
    clrwdt
    movlw code_hora2    ;Código para mostrar la HORA2
    xorwf character,W
    btfss STATUS,Z
    goto BUSX4411
    CLRF PCLATH
    call code_hora2_cod
    BSF PCLATH,3
    goto BUSXx5111
BUSX4411
    CLRF PCLATH
    CALL BUSCA_FECHACODE
    BSF PCLATH,3
    BTFSC FECHACODE
    goto BUSXx5111
    movlw temperatura    ;Código para mostrar la TMP
    xorwf character,W
    btfss STATUS,Z
    goto BUSX4111
    CLRF PCLATH
    call temperatura_cod
    BSF PCLATH,3
    goto BUSXx5111
BUSX4111
    movlw car_rotate
    xorwf character,W
    btfss STATUS,Z
    goto BUSXx511
    CLRF PCLATH
    call car_rotate_cod
    BSF PCLATH,3
    goto BUSXx5111
BUSXx511
    clrf pclath
    call code_velo    ;codigo de velocidad
    BSF PCLATH,3
    btfss codevelo
    goto BUSXx5111
    clrf pclath
    CALL set_codevelo_rot
    BSF PCLATH,3
    goto BUSXx5111
```

```

BUSXx5111
RETURN
ORG 0900H
ACTUALIZTMP
    clrwdt
    call ConfDS
    bsf STATUS,RP0
    clrf DatoRx_b2
    bcf STATUS,RP0
    call LeeT ;lee temperatura del ds1620 y la guarda en DatoRx_b2
    bsf STATUS,RP0
    movlw correccion_tmp
    movwf Conta4_b2
correc_tmp
    movf Conta4_b2,w
    xorlw .0
    btfsc status,z
    goto contitmp1
    decf Datorx_b2
    decf Conta4_b2
    goto correc_tmp
contitmp1
    bcf status,rp0
    call BinBCD
    bsf STATUS,RP0
    movf Decena_b2,w
    addlw 0x30 ;convierte en acii
    movwf Decena_b2

    movf unidad_b2,w
    addlw 0x30 ;convierte en acii
    movwf unidad_b2
    movf DIRVERTMP_H_b2,W
    xorlw 0x00
    btfss STATUS,Z
    goto GRATMP
    movf DIRVERTMP_L_b2,W
    xorlw 0x00
    btfsc STATUS,Z
    GOTO FIN_TMP ;No existe dir RAM para insertar la fecha
GRATMP
    clrwdt
    bsf STATUS,RP0
    movf DIRVERTMP_H_b2,W
    bcf STATUS,RP0
    movwf DIRH
    bsf STATUS,RP0
    movf DIRVERTMP_L_b2,W
    bcf STATUS,RP0
    movwf DIRL
    ;EN DATOX VIENE GURADADO EL VALOR DE LA
    bsf STATUS,RP0
    movf decena_b2,w
    bcf STATUS,RP0
    movwf datox
    movf pclath,w
    movwf pclathtemp
    clrf pclath
    call ESC_RAM
    movf pclathtemp,w
    movwf pclath
    CALL INC_DIRLHX
    bsf STATUS,RP0
    movf unidad_b2,w
    bcf STATUS,RP0
    movwf DATOX
    movf pclath,w
    movwf pclathtemp

```

59
90

```
    clrf   pclath
    call   ESC_RAM
    movf   pclathtemp,w
    movwf   pclath
    CALL   INC_DIRLHX
    movlw  .125
    movwf   DATOX
    movf   pclath,w
    movwf   pclathtemp
    clrf   pclath
    call   ESC_RAM
    movf   pclathtemp,w
    movwf   pclath
    CALL   INC_DIRLHX
    movlw  'C'
    movwf   DATOX
    movf   pclath,w
    movwf   pclathtemp
    clrf   pclath
    call   ESC_RAM
    movf   pclathtemp,w
    movwf   pclath
FIN_TMP
    bcf     STATUS,RP0
    RETURN
;*****
;*****
;*****
; ConfDS configura el DS1620 para comunicación con CPU, y conversiones continuas
; El byte de configuración es b'00001011', donde sólo los dos bits menos significativos
; son de interés. El bit 0 indica una sola conversión, y el 1 comunicación con CPU.
ConfDS
    clrwdt
    bsf     RSTT
    bsf     STATUS,RP0
    movlw  .8
    movwf   Conta4_b2
    movlw  0Ch ; Comando de escribir configuración
    movwf   DatoTx_b2
    bcf     STATUS,RP0
    call    OutBit

    bsf     STATUS,RP0
    movlw  .8
    movwf   Conta4_b2
    movlw  b'00001011' ; Registro de configuración
    movwf   DatoTx_b2
    bcf     STATUS,RP0
    call    OutBit
    bcf     RSTT
    retlw  0
; LeeT lee la temperatura desde el DS1620, con una resolución de 1°C. Envía el dato AAh
; (Leer temperatura), luego configura (DQ) como entrada, y recibe 9 bits enviados
; por el DS1620, dejando los 8 más significativos en DatoRx. Posteriormente vuelve a
; configurar DQ como salida.
;*****
;*****
LeeT
    CLRWDT
    bsf     RSTT
    bsf     STATUS,RP0
    movlw  .8
    movwf   Conta4_b2
    movlw  0AAh ; Comando de leer temperatura
    movwf   DatoTx_b2
    bcf     STATUS,RP0
    call    OutBit
    bsf     STATUS,RP0
```

60
59

```
bsf    TRIS_DQT
bcf    STATUS,RP0
    bsf    STATUS,RP0
movlw  .9
movwf  Conta4_b2
    bcf    STATUS,RP0
call   LeeBit
bsf    STATUS,RP0
bcf    TRIS_DQT
bcf    STATUS,RP0
bcf    RSTT
nop
nop
nop
nop
    clrwdt
bsf    RSTT
    bsf    STATUS,RP0
movlw  .8
movwf  Conta4_b2
movlw  0EEh ; Comando de iniciar conversión
movwf  DatoTx_b2
    bcf    STATUS,RP0
call   OutBit
bcf    RSTT
nop
nop
nop
retlw  40
;*****
;*****
; OutBit saca bits hacia el DS1620. Controla CLK y DQ. RST debe estar en bajo.
; Conta4 contiene el número de bits a sacar. DatoTx contiene el byte respectivo.
; Sale primero el LSB (Bit menos significativo).
OutBit
    bsf    STATUS,RP0
    rrf    DatoTx_b2,f
    bcf    STATUS,RP0
    bcf    DQT
    bcf    CLKT
    btfsc  STATUS,C
    bsf    DQT
    nop
    nop
    nop
    nop
    nop
    clrwdt
    bsf    CLKT
    nop
    nop
    nop
    bsf    STATUS,RP0
    decfsz Conta4_b2,f
    goto  OutBit
    bcf    STATUS,RP0
    retlw  0
;*****
;*****
; LeeBit lee bits desde el DS1620. Controla CLK y DQ. RST debe estar alto.
; Conta4 contiene el número de bits a leer. El dato se guarda en DatoRx.
; Llega primero el LSB (Bit menos significativo).
; 4 (DQT) debe estar definido como entrada.
LeeBit
    bcf    STATUS,RP0
    bcf    STATUS,C
    bcf    CLKT
    nop
```

```

nop
nop
CLRWDI
btfsc DQT
bsf STATUS,C
    bsf STATUS,RP0
rrf DatoRx_b2,f
    bcf STATUS,RP0
bsf CLKT
nop
nop
nop
    bsf STATUS,RP0
decfsz Conta4_b2,f
goto LeeBit
    bcf STATUS,RP0
retlw 0
; BinBCD fue tomado del "Embedded Control Handbook 1994/1995" - Microchip
; Convierte un byte (menor a 64H = 100 decimal) en dos dígitos BCD (Decena y Unidad)
; Se agregó la verificación del número, y la conversión del negativo.
; Se pueden mostrar números entre -9 y 99. Por encima de 99 sale en blanco.
; Por debajo de -9 se pierde el signo.
BinBCD
    CLRWDI
; bcf INTCON,GIE
    bsf STATUS,RP0
    clrf Decena_b2
    movfw DatoRx_b2
    movwf Unidad_b2
    movlw 7F
    subwf Unidad_b2,w
    btfsc STATUS,C
    call Negativ
    movlw 64
    subwf Unidad_b2,w
    btfsc STATUS,C
    goto Supera
gtenth
    movlw .10
    subwf Unidad_b2,w
    btfss STATUS,C
    goto over
    movwf Unidad_b2
    incf Decena_b2,f
    goto gtenth
over
    bsf INTCON,GIE
    bcf STATUS,RP0
    retlw 0
Supera
    movlw 0B
    movwf Unidad_b2
    movwf Decena_b2
    bsf INTCON,GIE
    bcf STATUS,RP0
    retlw 0
Negativ
    movlw 0FF
    xorwf Unidad_b2,f
    movlw 1
    addwf Unidad_b2,f
    movlw 0A
    movwf Decena_b2
    bsf INTCON,GIE
    retlw 0
;*****
;*****
INC_DIRLHX

```

```

clrwdt
incf DIRL,1
btfsc STATUS,Z
incf DIRH,1
RETURN
;*****
;   ORG    0C00H
INICIALIZACION
bsf CS
bsf WE    ;Lectura de RAM
bsf OER
bsf STATUS,RP0
movlw b'11111000'
movwf TRISC
bcf STATUS,RP0
bsf CS
bsf WE    ;Lectura de RAM
bsf OER
clrf INTCON
bcf STATUS,RP1
bsf STATUS,RP0
movlw .7
movwf ADCON1
movlw B'11111000' ;X X X RTC X STRB CLK DATA (PANEL)
movwf TRISA
movlw b'11111111'
movwf TRISB    ;Funciona como entrada (Leer RAM) y como salida (Escribir RAM).
movlw b'00000000'
movwf TRISD    ;STR94 DAT94 CLK94 OE94 CS OER WE DQT
movlw b'11011000'; RX TX HAB485 SDA SCL OEPWM CLKT RSTT
movwf TRISC
movlw b'00000000'
movwf TRISE
; bsf PIE1,TMR1IE ;Habilita interrupción debida a Timer1 en modo temporizador.
BCF PIE1,TMR1IE ;DESHabilita interrupción debida a Timer1 en modo temporizador.
bcf STATUS,RP0
clrwdt
bcf HabTx    ;Puerto RS 485 en modo RECEPCIÓN.
CLRF CURSOR
CLRF CURSOR_INI
CLRF NCVL
CLRF NCVH
CLRF NCSEP
CLRF NCSEPAUX
CLRF PCLATHTEMP
CLRF DESPOFFSETL
CLRF DESPOFFSETH
CLRF DIRECCIONR
CLRF DIRVERHR_H
CLRF DIRVERHR_L
CLRF POSIC_CAR
CLRF DESPLAZL
CLRF DESPLAZH
bcf aborto
clrf bandera0
clrf bandera1
clrf bandeMod
clrf bandera2
bsf status,rp0
clrf DIRH_TMP_b2
clrf DIRL_TMP_b2
clrf DATORX_b2
clrf DECENA_b2
clrf UNIDAD_b2
clrf DIRVERTMP_H_b2
clrf dirvertmp_l_b2
clrf conta4_b2
clrf datotx_b2

```

[illegible]

[illegible]

66
69

```
movwf pclath
movlw 0xFC
xorwf character,W
btfsc STATUS,Z
goto VERIFYDATA
;Chequea si Direccion actual (DIR) es Dirección final total de RAM (7FFFh):
movf DIRH,W
xorlw 0x7F
btfss STATUS,Z
goto CICLO_VERIF
movf DIRL,W
xorlw 0xFF
btfss STATUS,Z
goto CICLO_VERIF
call SETUPRAM2 ;No se encontró código de fin de mensaje, entonces graba mensaje por defecto.
VERIFYDATA
;Definido el intervalo de direcciones RAM donde se encuentran datos, verificar si estos datos
corresponden
;a códigos de caracteres ascii válidos o si son códigos de atributos. De no corresponder un dato con el
;código de un atributo o de un caracter ascii, se reemplaza el dato en dicha posición por el código ascii
;de un espacio en blanco.
clrf DIRH
clrf DIRL
CHEQDATA
clrwdt
movf pclath,w
movwf pclathtemp
clrf pclath
call LEE_RAM
movf pclathtemp,w
movwf pclath
CLRWDT
movlw 0xFE ;Código de texto NORMAL
xorwf character,W
btfss STATUS,Z
goto salatr11
goto SIGCHEQDATA
salatr11 clrwdt
movlw 0xFC ;Código de fin de mensaje
xorwf character,W
btfsc STATUS,Z
goto ACABOCHEQDATA
goto CHEQASCII2
SIGCHEQDATA
CALL INC_DIRLH2
goto CHEQDATA
CHEQASCII2
;Chequear si esta en el rango de caracteres válidos
movlw .32
movwf codascii
FINDASCII
CLRWDT
movf codascii,W
xorwf character,W
btfsc STATUS,Z
goto SIGCHEQDATA
movlw .156
xorwf codascii,W
btfsc STATUS,Z
goto INSERTA_SPACE
incf codascii,1
goto FINDASCII
INSERTA_SPACE
movlw 0x5B ;ESPACIO CORTO
movwf DATOX
movf pclath,w
movwf pclathtemp
clrf pclath
```

[illegible]

```

    bcf  status,z
    movlw 0x00
    xorwf DATOPC,W
    btfss STATUS,Z
    GOTO return_INT
    clrwdt
    btfss PIR1,RCIF
    goto $-1
    bcf  PIR1,RCIF
    movf RCREG,W
    movwf DATOPC
    bcf  STATUS,Z
    movlw INFO_IDEN ; codigo del informador xorwf DATOPC,W
    xorwf datopc,w
    btfsc STATUS,Z
    goto COMUNPC
    bcf  status,z
    movlw 0x00
    xorwf DATOPC,W
    btfss STATUS,Z
    GOTO return_INT
    clrwdt
    btfss PIR1,RCIF
    goto $-1
    bcf  PIR1,RCIF
    movf RCREG,W
    movwf DATOPC
    bcf  STATUS,Z
    movlw INFO_IDEN ; codigo del informador xorwf DATOPC,W
    xorwf datopc,w
    btfsc STATUS,Z
    goto COMUNPC
    bcf  status,z
return_INT
    CLRF PCLATH
    RETURN
; BCF OE
;espera_rx
; nop
; btfss PIR1,RCIF
; goto $-1
; bcf  PIR1,RCIF
; movf RCREG,W
; clrwdt
; goto espera_rx
; GOTO $
;Aquí se establece la comuincación con el PC si el código del Informador es correcto.
;Espera código de tarea arealizar:
COMUNPC
    BCF  OE
;Espera código de tarea:
    bcf  PIR1,RCIF
    clrwdt
    btfss PIR1,RCIF
    goto $-1
    bcf  PIR1,RCIF
    movf RCREG,W
    movwf DATOPC
    movlw 0x01 ;Código de tarea para grabar datos en RAM
    xorwf DATOPC,W
    btfsc STATUS,Z
    goto GRABARAM
    movlw 0x02 ;Código de tarea para configurar el reloj de tiempo real.
    xorwf DATOPC,W
    btfsc STATUS,Z
    goto GRABARELOJ
    movlw 0x03 ;Código de tarea para configurar el Informador.
    xorwf DATOPC,W

```

69
100

```
btfsc STATUS,Z
goto CONFIG_INFO
NOP
BCF OE
espera_commpc
btfss PIR1,RCIF
goto $-1
bcf PIR1,RCIF
movf RCREG,W
clrwdt
goto espera_commpc
GOTO $
CONFIG_INFO
clrwdt
movlw 0x20
movwf FSR
movlw .6
movwf cuantos
RECICONFIG
;clrwdt
btfss PIR1,RCIF
goto $-1
bcf PIR1,RCIF
movf RCREG,W
movwf INDF
incf FSR,1
decfsz cuantos,1
goto RECICONFIG
movlw 0x20
movwf FSR
bsf STATUS,RP1
bcf STATUS,RP0 ;Banco 2 de RAM
clrf EEADR ;Posición inicial en memoria EEPROM para almacenar los datos.
bcf STATUS,RP1 ;Banco 0 de RAM
movlw .6 ;Número de datos a grabar.
movwf cuantos
call Esc_EEP
bcf OE
espera_config
btfss PIR1,RCIF
goto $-1
bcf PIR1,RCIF
movf RCREG,W
clrwdt
goto espera_config
GOTO $
;-----
;-----
GRABARELOJ
clrwdt
movlw 0x20
movwf FSR
movlw .7
movwf cuantos
RECIDATRELOJ
btfss PIR1,RCIF
goto $-1
bcf PIR1,RCIF
movf RCREG,W
movwf INDF
incf FSR,1
decfsz cuantos,1
goto RECIDATRELOJ
movlw 0x20
movwf FSR
movlw .7 ;Número de datos a grabar.
movwf cuantos
call ESCDS1307
```

70
69

```

BCF    OE
espera_rxrelog
    btfss PIR1,RCIF
    goto  $-1
    bcf    PIR1,RCIF
    movf  RCREG,W
    clrwdt
    goto  espera_rxrelog
GOTO  $
;-----
GRABARAM
    clrf  DIRL
    clrf  DIRH
    call  clear_retard
    bcf   sirx_retard
    clrwdt
SIGGRAB
    incf  rx_retardl,1    ;espera que da un tiempo límite para la recepcion del
    NOP
    NOP
    NOP
    btfss status,z       ;siguiente dato. si se llega al limite rx_retardh=0
    goto  cont_siggrab    ;entonces se graba un final de caracter 0xFC y se reinicia
    incf  rx_retardh,1
    btfss status,z
    goto  cont_siggrab
    incf  rx_retardhh,1
    btfss status,z
    goto  cont_siggrab
    movlw 0xFC
    bsf   sirx_retard
    goto  cont_sigrab2
cont_siggrab
    btfss PIR1,RCIF
    goto  SIGGRAB
    bcf   PIR1,RCIF
    movf  RCREG,W
cont_sigrab2
    clrwdt
    movwf DATOPC
    movwf DATOX
    clrwdt
    call  clear_retard
    movf  pclath,w
    movwf pclathtemp
    clrf  pclath
    call  ESC_RAM
    movf  pclathtemp,w
    movwf pclath
    CALL  INC_DIRLH2
    movlw 0xFC
    xorwf DATOPC,W
    btfss STATUS,Z       ;dato recibido es fin de mensaje?
    goto  SIGGRAB        ;No!
    call  respuesta      ;ENVIA RESPUESTA PARA CONFIRMACIÓN DE TX
    btfsc sirx_retard
    goto  SalInterr
espera_fin
    btfss PIR1,RCIF
    goto  $-1
    bcf   PIR1,RCIF
    movf  RCREG,W
    btfss PIR1,RCIF
    goto  $-1
    bcf   PIR1,RCIF
    movf  RCREG,W
    clrwdt
    goto  espera_fin

```

21
70

```

;fin_rxx
; bsf STATUS,RP0
; bsf PIE1,RCIE
; bcf STATUS,RP0
;return
; clrf pclath
; goto INICIO
SalInterr
GOTO $
; retfie
; clrf pclath
; BSF STROBE
; return
clear_retard
clrf rx_retardl
clrf rx_retardh
movlw .150
movwf rx_retardhh
return
;*****
SETUPRAM2
clrwdt
bcf colvacias ;Si hay columnas vacías.
clrf DIRH
movlw .1
movwf DIRL
movlw '!'
movwf DATOX
movf pclath,w
movwf pclathtemp
clrf pclath
call ESC_RAM
movf pclathtemp,w
movwf pclath
CALL INC_DIRLH2
movlw 0xFC
movwf DATOX
movf pclath,w
movwf pclathtemp
clrf pclath
call ESC_RAM
movf pclathtemp,w
movwf pclath
; RAM 0001H = !
;Establece número de columnas vacías [NCT - 2]: 0002H = FF
bcf STATUS,C
movlw .5
subwf NCTL,W
movwf NCVL
btfss STATUS,Z
goto MIRANEGXX3
movlw .0
xorwf NCTH,W
btfss STATUS,Z
goto SALRESTA223
clrf NCVH
bsf colvacias ;No hay columnas vacías!!
goto ESTATICO3
MIRANEGXX3
btfsc STATUS,C ;resultado de resta anterior negativo?
goto SALRESTA223 ;No
decf NCTH,W ;Sí
movwf NCVH
goto ESTATICO3
SALRESTA223
movf NCTH,W
movwf NCVH
ESTATICO3

```

72
71

```
bcf modoRotate
clrf DirIniH
clrf DirIniL
clrf cursor_ini
return
;*****
INC_DIRLH2
    incf DIRL,1
    btfsc STATUS,Z
    incf DIRH,1
    RETURN

;*****
;*****
;ESC_RAM clrwdt          ;ESCRIBE EN RAM EXTERNA 'DATOX'. PARTE ALTA 'DIRH' ES
DIRECCIONADA DE FORMA SERIAL POR UN 4094
; bcf STR94              ;EN LA RUTINA ENVIDIRH Y LA PARTE BAJA 'DIRL' ES DIRECCIONADA POR
EL PUERTO D
; movf DIRH,W
; call ENVDIRH
; bsf STR94
; movf DIRL,W
; movwf PORTD
; bsf STATUS,RP0
; movlw 0x00
; movwf TRISB
; bcf STATUS,RP0
; movf DATOX,W
; movwf PORTB
; bcf WE      ;escritura en RAM
; bcf OER     ;Entrada de datos a RAM habilitada
; bcf CS      ;Selecciona RAM
; nop
; nop
; bsf CS      ;RAM inhabilitada
; bsf WE      ;Lectura de RAM
; bsf OER     ;Entrada de datos a RAM deshabilitada
; return
;*****
;lee_ram2
; movlw LOW FRASE
; addwf DIRL,W
; movwf caraux
; movlw HIGH FRASE
; addwf DIRH,W
; movwf PCLATHtemp
; movf PCLATHtemp,W
; clrf pclath
; bsf PCLATH,3
; bsf PCLATH,4
; call FRASE      ;Se encuentra en página 2 de memoria de programa.
; movwf CHARACTER
; clrf pclath
; bsf PCLATH,3
; bsf PCLATH,2
; return
;LEE_RAM2I
; clrwdt
; bcf STR94
; movf DIRH,W
; call ENVDIRH
; bsf STR94
; movf DIRL,W
; movwf PORTD
; bsf STATUS,RP0
; movlw 0xFC
; movwf TRISB
; bcf STATUS,RP0
```

73
72

```

; bsf WE ;Lectura de RAM
; bcf OER ;Salida de datos de RAM habilitada
; bcf CS ;Selecciona RAM
; movf PORTB,W
; movwf caracter
; nop
; bsf CS ;RAM inhabilitada
; bsf OER ;Entrada de datos a RAM deshabilitada
; return
;
;*****
;ENVDIRH
;movwf DIR_RAM_H
; movlw .8
; movwf contbits
;TX4094b bcf CLK94
; clrwdt
; bcf DAT94 ;Coloca en cero línea de datos del 4094
; btfsc DIR_RAM_H,7
; bsf DAT94 ;Coloca en alto línea de datos del 4094
; nop
; clrwdt
; bsf CLK94 ;Pulso de reloj para los registros 4094
; rlf DIR_RAM_H,1 ;Rota byte que se está enviando (DATOCOL)
; decfsz contbits,1
; goto TX4094b
; return
;*****
;***** RUTINAS DEL BUS I2C *****
;*****
;-----
LEEDS1307 ;Lee la hora actual y la fecha del reloj.
clrwdt
movlw .6
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf an_o
movlw .5
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf mes
movlw .4
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf dia_mes
movlw .3
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf dia_sem
movlw .2
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf hora
movlw .1
movwf DIRECCIONR
call LEER_R
movf RECEP,W
movwf minutos
clrwdt
clrf pclath
return
;*****
;
LEER_R call BIT_INICIO ;Lee 1 byte del reloj serial DS1307.

```

74
73

```

nop
nop
movlw b'11010000' ;Código de dispositivo en el bus I2C. [bcf  DISPTIVO,0]
movwf TRANSM
call ENVIA_BYTE
movf DIRECCIONR,W
movwf TRANSM
call ENVIA_BYTE
nop
nop
clrwdt
call BIT_INICIO
nop
nop
movlw b'11010001' ;Código de dispositivo en el bus I2C. [bsf  DISPTIVO,0]>Lectura.
movwf TRANSM
clrwdt
call ENVIA_BYTE
nop
call RECIBE_BYTE
bsf  EEPROM,BIT_OUT
clrwdt
call SACA_BIT
call BIT_PARO
return
,*****
,
```

BIT_INICIO ;Genera bit de inicio (START).

```

;clrwdt
bsf STATUS,RP0
movlw b'11000000'
movwf TRISC
bcf STATUS,RP0
bcf  SCL
bsf  SDA
nop
bsf  SCL
nop
nop
nop
nop
nop
nop
bcf  SDA
nop
nop
nop
nop
nop
nop
bcf  SCL
nop
nop
return
,*****
,
```

BIT_PARO ;Genera el bit de parada (STOP).

```

;clrwdt
bsf STATUS,RP0
movlw b'11000000'
movwf TRISC
bcf STATUS,RP0
bcf  SDA
nop
nop
nop
bsf  SCL
nop
nop
nop
bsf  SDA
nop
,
```

75
74

```

nop
bcf SCL
nop
nop
nop
return
;-----
ESCDS1307 ;Escribe la hora y la fecha al reloj.
    movlw .6
    movwf DIRECCIONR
NEXTESCDS1307
    CLRWDT
    movf INDF,W
    bsf STATUS,RP0
    movwf DATO_b2
    bcf STATUS,RP0
    call ESCRIBIR_R
    decf DIRECCIONR,1
    incf FSR,1
    decfsz cuantos,1
    goto NEXTESCDS1307
    clrf DIRECCIONR
    bsf STATUS,RP0
    clrf DATO_b2
    bcf STATUS,RP0
    CLRWDT
    call ESCRIBIR_R
    return
;*****
;-----
ESCRIBIR_R ;Escribe 1 byte en el reloj serial DS1307.
    call BIT_INICIO
    nop
    nop
    movlw b'11010000' ;Código de dispositivo en el bus I²C. [bcf DISPTIVO,0]
    movwf TRANSM
    call ENVIA_BYTE
    movf DIRECCIONR,W
    movwf TRANSM
    call ENVIA_BYTE
    nop
    nop
;clrwdt
    bsf STATUS,RP0
    movf DATO_b2,W
    bcf STATUS,RP0
    movwf TRANSM
    call ENVIA_BYTE
    call BIT_PARO
    return
;*****
;-----
ENVIA_BYTE ;Envía un byte por el bus I²C al dispositivo esclavo.
    movlw .8
    movwf cuenta
TXLP ;clrwdt
    bcf EEPROM,BIT_OUT
    btfsc TRANSM,7
    bsf EEPROM,BIT_OUT
    call SACA_BIT
    rlf TRANSM,1
    decfsz cuenta,1
    goto TXLP
    call LEE_BIT ;lee (espera) reconocimiento del dispositivo.
    return
RECIBE_BYTE ;Recibe un byte por el bus I²C del dispositivo esclavo.
    clrf RECEP
    movlw .8
    movwf cuenta
```

76
75

```

    bcf STATUS,0
RXLP    clrwdt
    rlf RECEP,1
    call LEE_BIT
    btfsc EEPROM,BIT_IN
    bsf RECEP,0
    decfsz cuenta,1
    goto RXLP
    return
;*****
LEE_BIT clrwdt
    bsf EEPROM,BIT_IN ;Lee un bit en la línea de datos (SDA).
    bsf STATUS,RP0
    movlw b'11010000'
    movwf TRISC
    bcf STATUS,RP0
    bsf SCL
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    btfss SDA
    bcf EEPROM,BIT_IN
    bcf SCL
    return
;*****
SACA_BIT ;Saca un bit por la línea de datos (SDA).
    clrwdt
    bsf STATUS,RP0
    movlw b'11000000'
    movwf TRISC
    bcf STATUS,RP0
    bsf SDA
    btfss EEPROM,BIT_OUT
    bcf SDA
    nop
    nop
    bsf SCL
    nop
    nop
    nop
    nop
    bcf SCL
    return
;*****
;-----
Lee_EEP2
    clrwdt
    bsf STATUS,RP1
    bsf STATUS,RP0 ;Banco 3 de RAM
    bcf EECON1,EEPGD ;PROGRAM/DATA SELECT BIT; ACCESO A LA MEMORIA EEPROM DE
DATOS
    bsf EECON1,RD ;INICIA LECTURA DE LA MEMORIA EEPROM DE DATOS
    bcf STATUS,RP0 ;Banco 2 de RAM
    movf EEDATA,w
    bcf STATUS,RP1 ;Banco 0 de RAM
    movwf INDF
    bsf STATUS,RP1
    bcf STATUS,RP0 ;Banco 2 de RAM
    incf EEADR,1
    bcf STATUS,RP1 ;Banco 0 de RAM
    incf FSR,1
    decfsz cuantos,1
    goto Lee_EEP2

```

77
76

```
return
;*****
;*****
respuesta
    clrf temp_l           ;Reg temporales guardan CRC temp ->CRC
    clrf temp_h
    clrf DIRH
    clrf DIRL
    clrwdt
    call retardo_rtx
    bsf habtx
genera_crc
    movf pclath,w
    movwf pclathtemp
    clrf pclath
    call LEE_RAM
    movf pclathtemp,w     ;CRC =
    movwf pclath
;PRUEBABABABA
; MOVF CHARACTER,W
; bcf PIR1,TXIF          ;Restaura el flag del transmisor
; movwf TXREG            ;Almacena el byte a transmitir
; bsf STATUS,RP0
pruebaba
; btfss TXSTA,TRMT       ;Byte transmitido ??
; goto pruebaba          ;No, esperar
; bcf STATUS,RP0
;HKJHKJHKJH
    swapf caracter,w      ;w = caracter swaped
    andlw 0x0F
    addwf temp_l,1        ;crc = crc + w
    btfsc STATUS,C
    incf temp_h,1
    CALL INC_DIRLH2
    clrwdt
    movlw 0xFC
    xorwf caracter,W      ;verifica si es fin de caracter
    btfss STATUS,Z
    goto genera_crc
envia_crc
    movlw 0x02            ;codigo que identifica la tarjeta microcinta
    bcf PIR1,TXIF         ;Restaura el flag del transmisor
    movwf TXREG           ;Almacena el byte a transmitir
    bsf STATUS,RP0
Tx_Dato_Wait0
    btfss TXSTA,TRMT      ;Byte transmitido ??
    goto Tx_Dato_Wait0    ;No, esperar
    bcf STATUS,RP0
    movf temp_l,w
    bcf PIR1,TXIF         ;Restaura el flag del transmisor
    movwf TXREG           ;Almacena el byte a transmitir
    bsf STATUS,RP0
Tx_Dato_Wait1
    btfss TXSTA,TRMT      ;Byte transmitido ??
    goto Tx_Dato_Wait1    ;No, esperar
    bcf STATUS,RP0
    movf temp_h,w
    bcf PIR1,TXIF         ;Restaura el flag del transmisor
    movwf TXREG           ;Almacena el byte a transmitir
    bsf STATUS,RP0
Tx_Dato_Wait2
    btfss TXSTA,TRMT      ;Byte transmitido ??
    goto Tx_Dato_Wait2    ;No, esperar
    bcf STATUS,RP0
    clrwdt
envia_columnas
    movf NCTL,w
    bcf PIR1,TXIF         ;Restaura el flag del transmisor
```

28
77

```

movwf TXREG          ;Almacena el byte a transmitir
bsf STATUS,RP0
Tx_Dato_Wait3
btfss TXSTA,TRMT      ;Byte transmitido ??
goto Tx_Dato_Wait3    ;No, esperar
bcf STATUS,RP0
movf NCTH,w
bcf PIR1,TXIF         ;Restaura el flag del transmisor
movwf TXREG          ;Almacena el byte a transmitir
bsf STATUS,RP0
Tx_Dato_Wait4
btfss TXSTA,TRMT      ;Byte transmitido ??
goto Tx_Dato_Wait4    ;No, esperar
bcf STATUS,RP0
fin_crc
movlw 0xFC
bcf PIR1,TXIF         ;Restaura el flag del transmisor
movwf TXREG          ;Almacena el byte a transmitir
bsf STATUS,RP0
Tx_Dato_Wait5
btfss TXSTA,TRMT      ;Byte transmitido ??
goto Tx_Dato_Wait5    ;No, esperar
bcf STATUS,RP0
bcf habtx             ;deshabilita tx por RS485
return
;
;*****
retardo_rtx
clrf rx_retardl
clrf rx_retardh
rt_rx
movlw .220            ;APROX 5 SEGUNDOS. A MENOR W MAYOR RETARDO
rt_wr
clrwdt
addlw .1
btfss status,z
goto rt_wr
incf rx_retardl,1     ;espera que da un tiempo límite para la recepcion del
btfss status,z        ;siguiente dato. si se llega al limite rx_retardh=0
goto rt_rx            ;entonces se graba un final de caracter 0xFC y se reinicia
incf rx_retardh,1
btfss status,z
goto rt_rx
return
ORG 1800H
FRASE
movwf PCLATH
movf caraux,W
addwf PCL,1
; DT 0x5b,0x84
; DT "PUBLIK INFO"
; DT PARAVERIFICAR EL FUNCIONAMIENTO DEL
EFECTO HOLD CUANDO SE BLOQUEA PARAMENSAJES LARGOS"
;0x04,0x01,
; DT 0x5B,0xFE,0x9A,0x54,0x4D,0x50,0xFE,0x20,0xFC,0xFC,0xFC,0xFC,0xFC
; dt 0x5B,0x68,0x6F,0x72,0x61,0xFE,0x98,0x48,0x4F,0x52,0x41,0x4E,0xFE
; dt
0x98,0x54,0x45,0x4D,0x50,0xFE,0x97,0x48,0x4F,0x52,0x41,0x4E,0xFE,0x20,0xFC,0xFC,0xFC
; dt 0x5B,0x68,0x6F,0x72,0x61,0xFE,0x98,0x86,0x48,0x4F,0x52,0x41,0x4E,0x85,0xFE
; dt 0x98,0x54,0x45,0x4D,0x50,0xFE,0x97,0x46,0x45,0x43,0x48,0x41,0x31,0x20,0x48
; dt 0x4F,0x52,0x41,0x4E,0xFE,0x9A,0x54,0x45,0x4D,0x50,0xFE,0x98,0x48,0x4F,0x52
; dt 0x41,0x4E,0xFE,0x20,0xFC,0xFC,0xFC
dt
0x5B,0xFE,0x9B,0x5B,0x8C,0x5B,0x5B,0x5B,0x5B,0x5B,0xfc,0x5B,0x5B,0x5B,0x20,0xFE,0x97,0x5B,0x8B
,0x5B,0x5B,0x5B
dt
0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x5B,0x20,0xFE,0x20,0xFC,0xFC,0xFC
C
; dt 0x5B,0xFE,0x98,0x86,0x5B,0x88,0x5B,0x5B,0x5B,0x5B,0x5B,0x20,0x85,0xFE,0x98

```

79
70

```
; dt 0x45,0x73,0x74,0x65,0xFE,0x98,0x61,0x7C,0x6F,0xFE,0x97,0x73,0x65,0x20,0x62
; dt 0x65,0x6E,0x65,0x66,0x69,0x63,0x69,0x61,0x72,0x60,0x6E,0x20,0x31,0x30,0x30
; dt 0x65,0x6E,0x74,0x61,0x64,0x6F,0x20,0x6C,0x61,0x20,0x63,0x6F,0x62,0x65,0x72
; dt 0x74,0x75,0x72,0x61,0x20,0x65,0x64,0x75,0x63,0x61,0x74,0x69,0x76,0x61,0xFE
; dt 0x98,0x65,0x6E,0xFE,0x98,0x32,0x2E,0x35,0x33,0x39,0x20,0xFE,0x97,0x6E,0x75
; dt 0x65,0x76,0x6F,0x73,0x20,0x61,0x6C,0x75,0x6D,0x6E,0x6F,0x73,0xFE,0x20,0xFC
; dt 0xFC,0xFC
;   DT   0x88
;   DT   "023:23O"
;   dt   0x9A   ;   ,0x09,0xFE,0xFC
;   DT   "Hora: "
;   DT   0x89,0x45,0xFE,0xFC
; DT "HO"
;   DT   0xFE,0x17
;   DT "TU"
;   DT   0xFE,0xFC
;   DT   "a b c d e f g h i j k l m n"
;   DT   "o p q r s t u v w x y z a b c"
;   DT   "d e f g h i j k l m n o p q"
;   DT   "r s t u v w x y z"
END
```

80
74

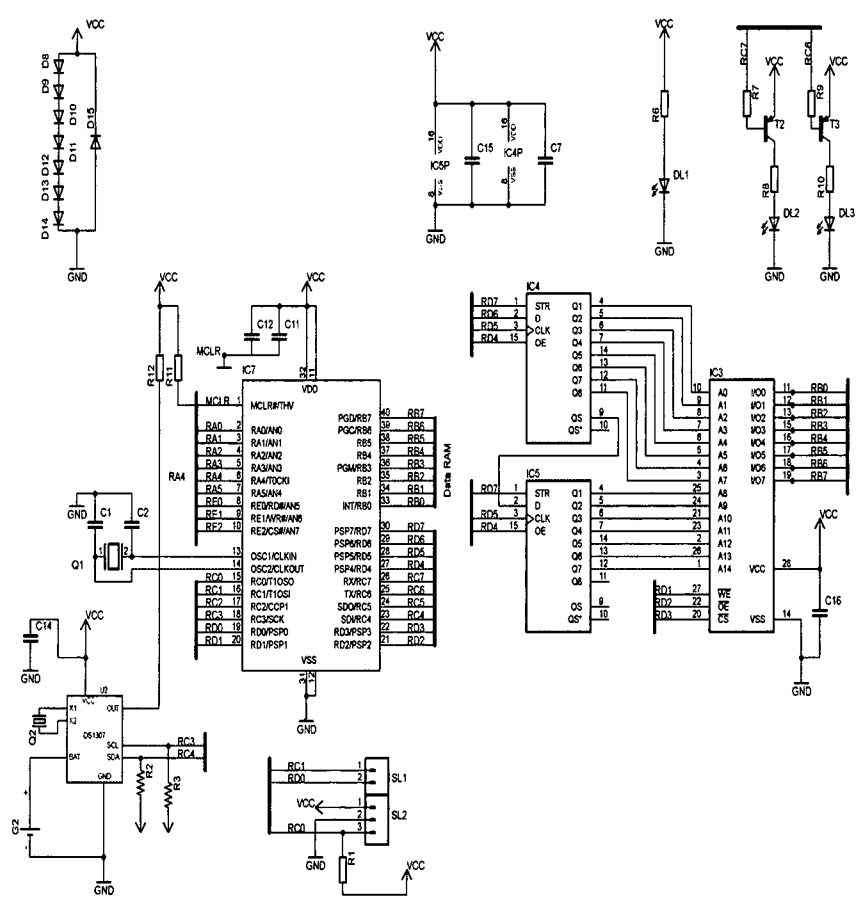


Diagrama 1. - Etapa de Control

81
80

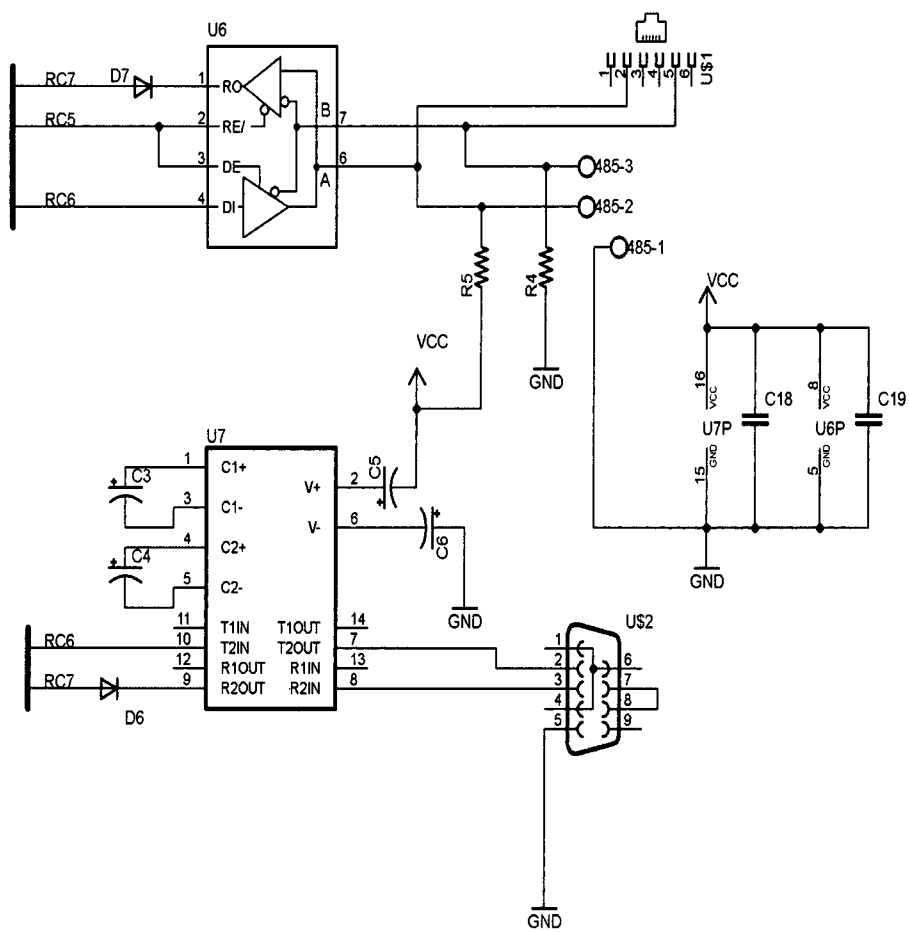


Diagrama 2. Etapa de Comunicación

82
21

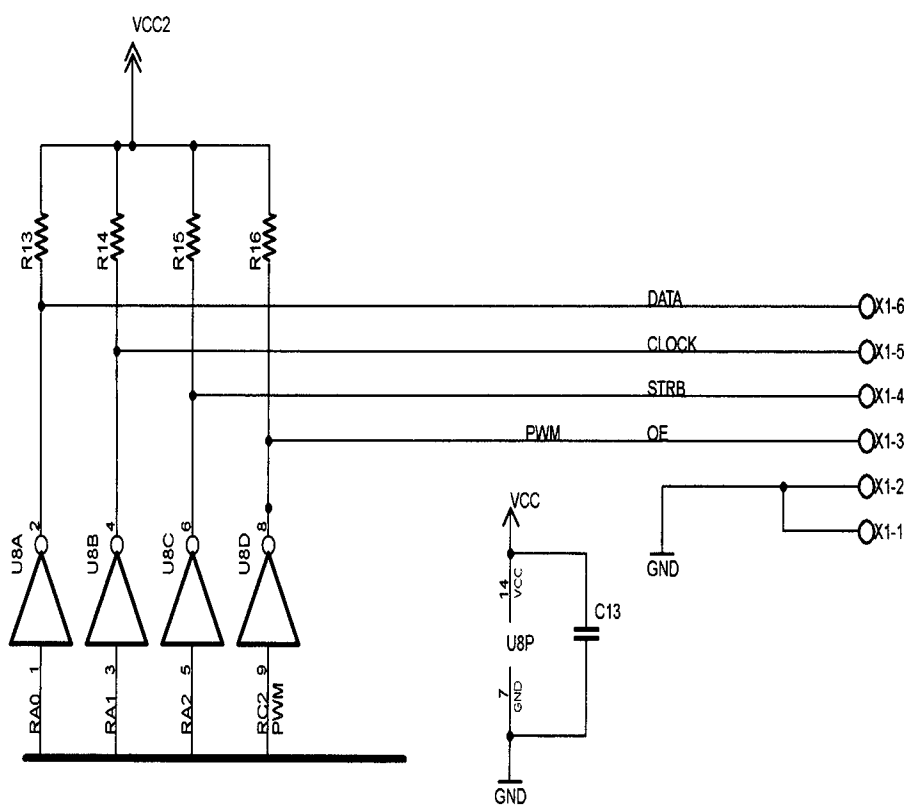


Diagrama 3. Etapa a la Salida de Potencia

83
02

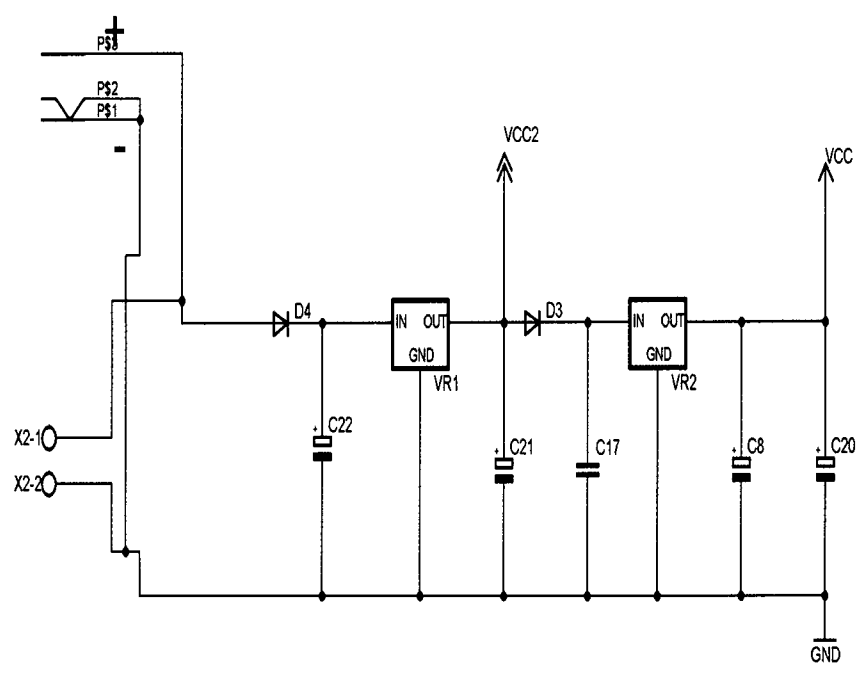


Diagrama 4. Fuentes de Alimentación

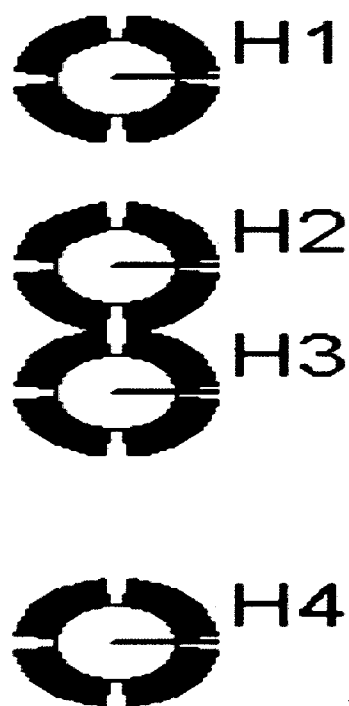
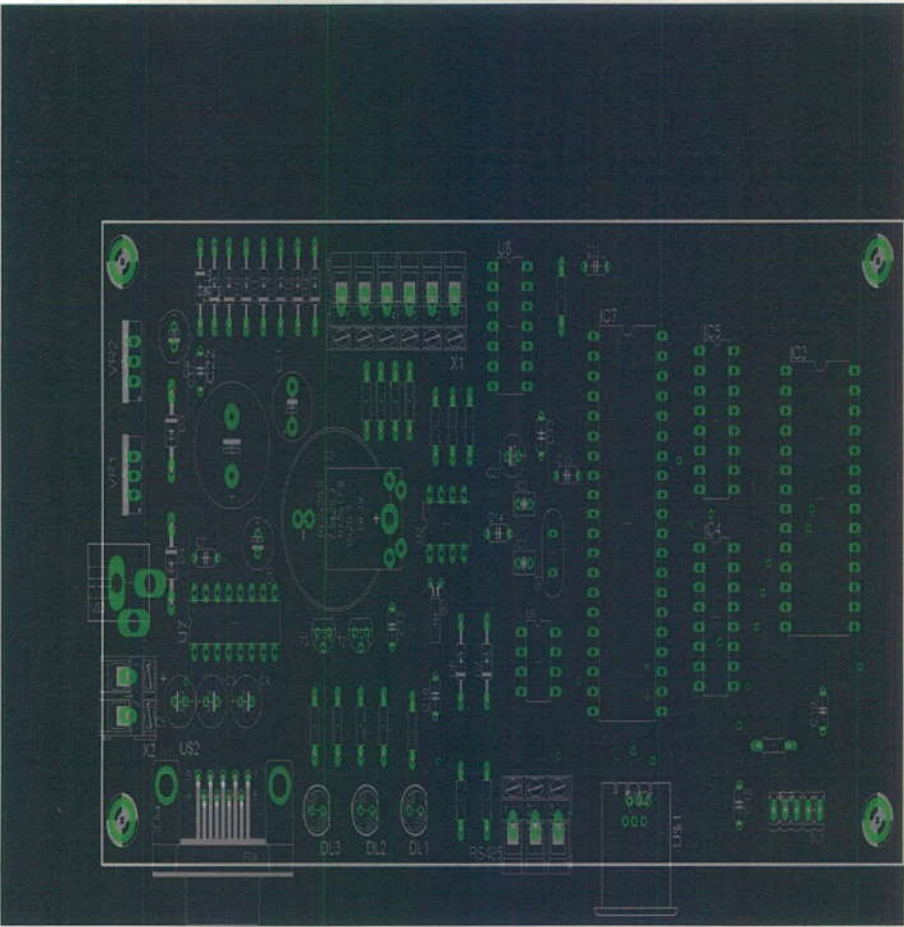


Diagrama 5. Hoyos para fijación de Tarjeta

85

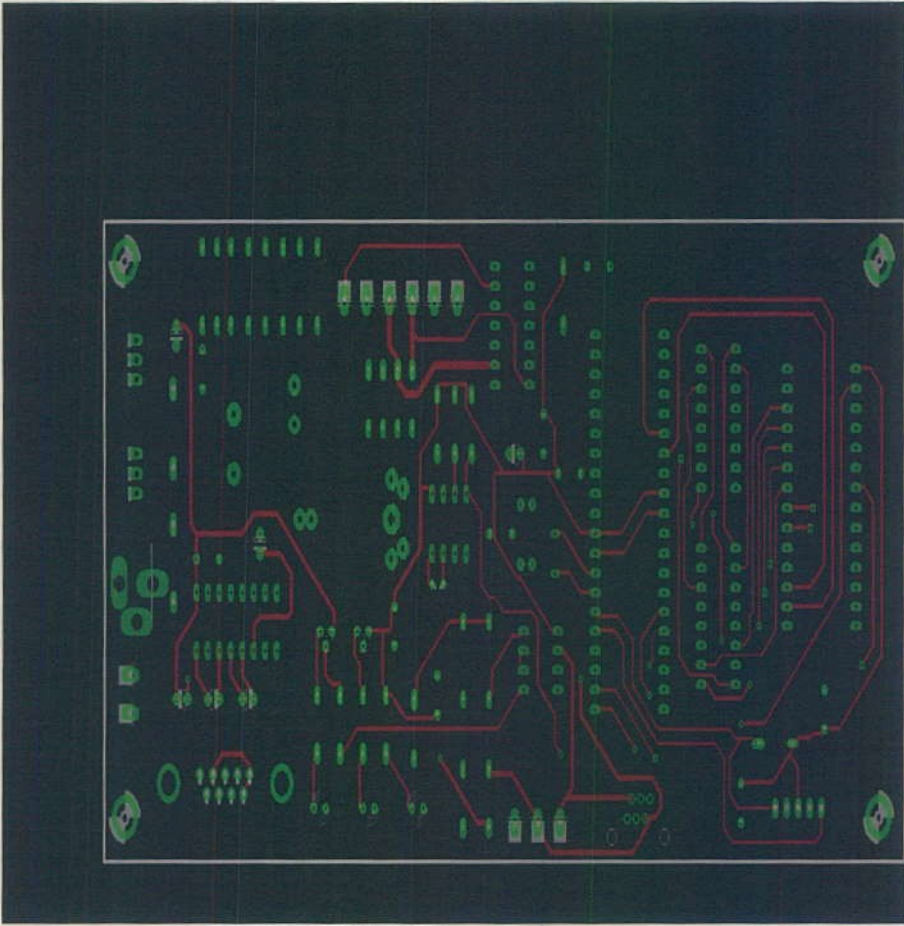
04



Grafica 1. Impreso de la Tarjeta MicroPublik - Componentes

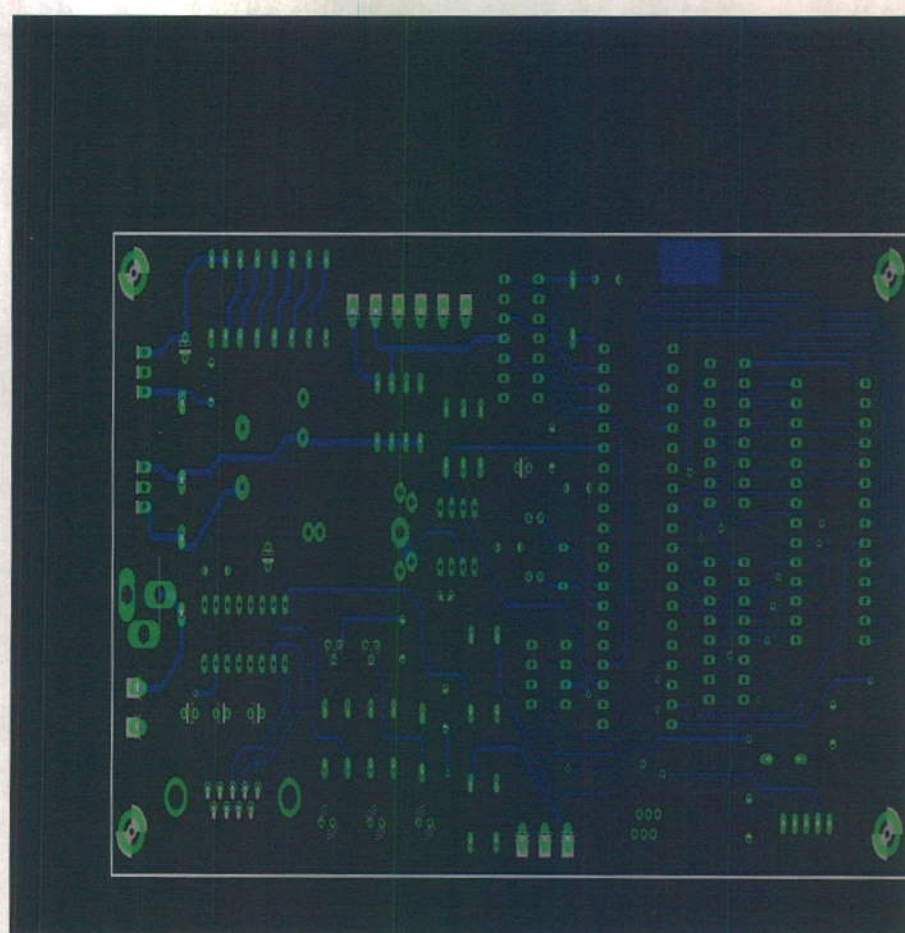
86

89



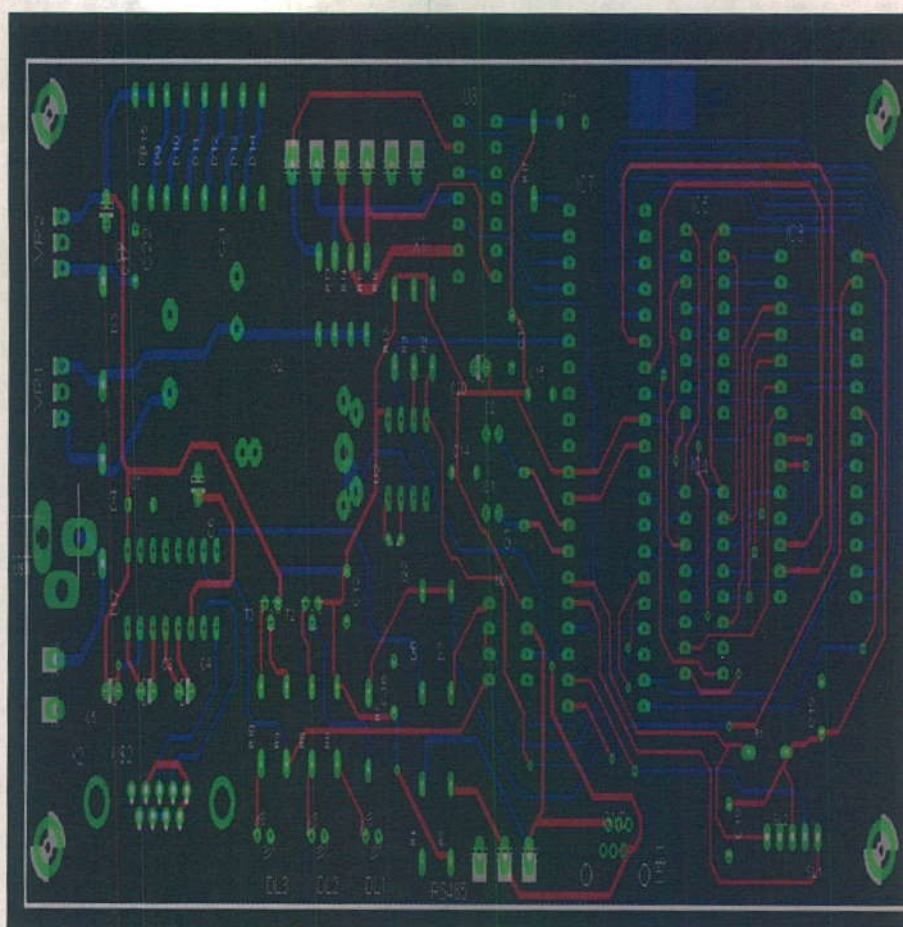
Grafica 2. Impreso de la Tarjeta MicroPublik - Cara Superior

87
86

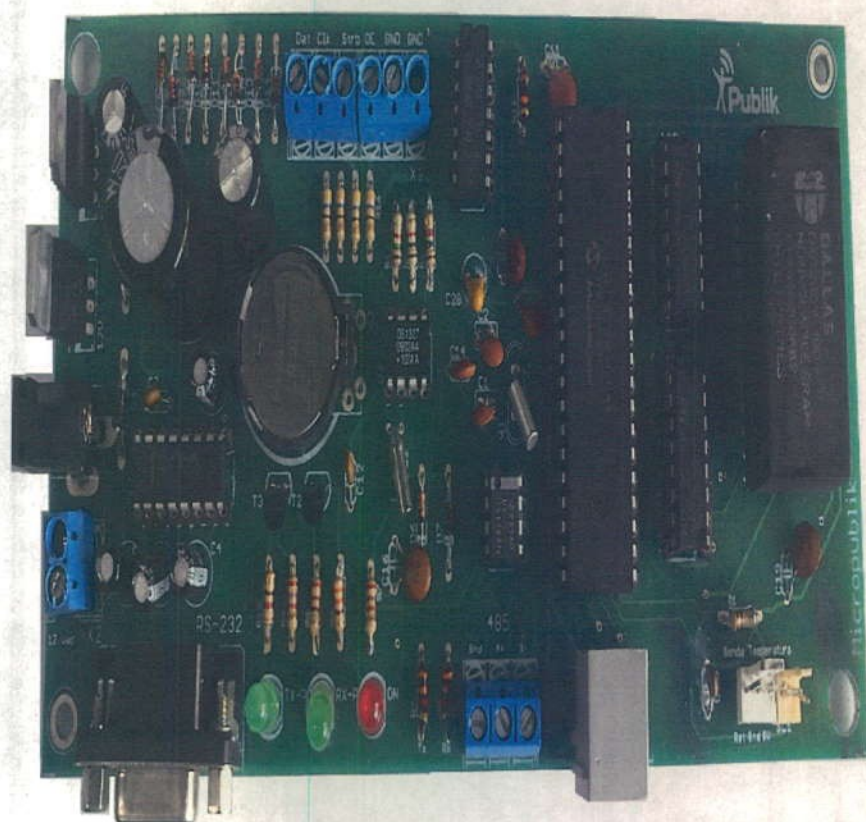


Grafica 3. Impreso de la Tarjeta MicroPublik - Cara Inferior

88
87



Grafica 4. Impreso de la Tarjeta MicroPublik - Componentes y
ambas caras



Grafica 5. Tarjeta MicroPublik v2.0

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

NIT : 800.176.089-2

-/-

Industria y Comercio
SUPERINTENDENCIA

RECIBO DE CAJA

No. 14 - 0031063

Bogotá D.C., Marzo 20 de 2014 - 12:47:38

RECIBIDO DE : PUBLIK TECNOLOGIAS INFORMACION COMUNICACIONES Y MEDIO AMBIENTE NI 891.902.435

RE

*** Soporte del Pago ***

TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PAGO	VR. PAGO
CONSIGNACION	BANCO DE BOGOTA	062754387	649753193	18/03/2014	278.000.00
CONSIGNACION	BANCO DE BOGOTA	062754387	649753204	18/03/2014	287.000.00

*** Conceptos Pagados ***

CANT. RENTISTICO	CONCEPTO	Vr.UNDITARIO	Vr.CONCEPTO
1 50005-01-01 SOLICITUDES	5 TRAMITES DE SOL. DE MODELO DE UTILIDAD	278.000.00	278.000.00
			\$278.000.00

SON: **DOSCIENTOS SETENTA Y OCHO MIL PESOS MONEDA CORRIENTE***

Responsable:

Recibo de Caja Aplicado al Expediente No. _____

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO



No. 14-060246- -00000-0000

Fecha: 2014-03-20 13:45:34 Dep. 2020 DIR.NUEVASCR
Tra. 3 MODELO Eve: 1 REGDEPOSITO
Act. 411 PRESENTACION Folios: 90

Sede Centro: Carrera 13 No. 27 - 00 Pisos 3,4,5 y 10 Bogotá, D.C.- Colombia

Web: www.sic.gov.co e-mail: info@sic.gov.co Conmutador: (571) 5870000 Fax: (571) 5870284 Línea: 018000-910165 Call Center: (571) 6513240

, Marzo 20 de 2014 - 12:47:39



No. 14-060246- -00000-0000

Fecha: 2014-03-20 13:45:34 Dep. 2020 DIR.NUEVASCR
Tra. 3 MODELO Eve: 1 REGDEPOSITO
Act. 411 PRESENTACION Folios: 90

PATENTE DE INVENCION ☐

MODELO ☐

- ☐ Indicación que se solicita una patente.
 - ☐ Datos de identificación del solicitante o de la persona que presenta la solicitud
 - ☐ Descripción de la invención
 - ☐ Dibujos de ser estos pertinentes
 - ☐ Comprobante de pago de las tasas establecidas (De ser el caso formato de descuento)
- Completa ☐ Incompleta ☐

PATENTE DE INVENCION PCT ☐

MODELO DE UTILIDAD PCT ☐

Art.33 Decisión 486/00, Circular Única

- ☐ Indicación que se solicita una PCT
 - ☐ Copia de la solicitud en español, tal como fue presentada inicialmente (capítulo descriptivo, reivindicatorio, resumen)
 - ☐ Dibujos de ser estos pertinentes
 - ☐ Comprobante de pago de las tasas establecidas (de ser el caso formato de descuento)
- Completa ☐ Incompleta ☐

DISEÑO INDUSTRIAL ☐

(Art. 119 Decisión 486/00)

- ☐ Indicación que se solicita Diseño industrial
 - ☐ Datos de identificación del solicitante o de la persona que presenta la solicitud
 - ☐ Representación gráfica y fotográfica del Diseño industrial o muestra del material que incorpora el diseño
 - ☐ Comprobante de pago de las tasas establecidas
- Completa ☐ Incompleta ☐

ESQUEMA DE TRAZADO ☐

(Art. 92 Decisión 486/00)

- ☐ Indicación que se solicita un esquema de trazado
 - ☐ Datos de identificación del solicitante o de la persona que presenta la solicitud
 - ☐ Representación gráfica de un esquema de trazado
 - ☐ Comprobante de pago de las tasas establecidas
- Completa ☐ Incompleta ☐