

RT-C2

Recolector de tickets

CARACTERÍSTICAS DEL HARDWARE

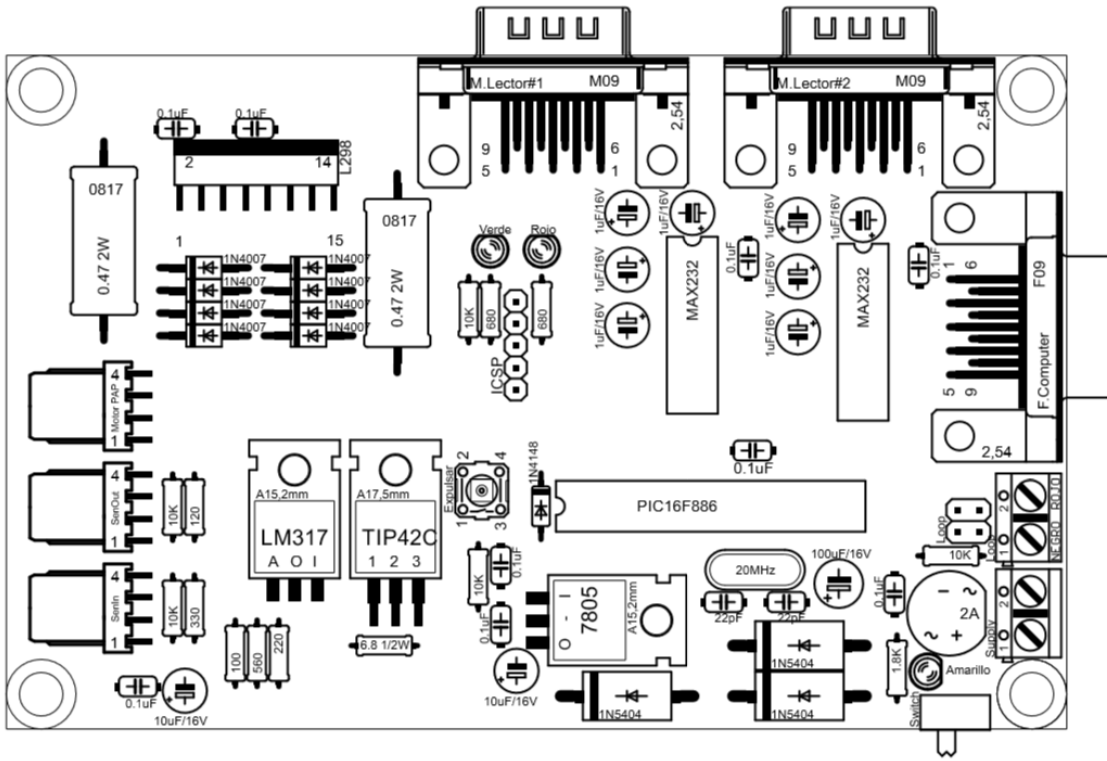
- Cuenta con un motor de pasos como dispositivo actuador encargado de posicionar el ticket en el área de lectura, de almacenar el ticket dentro del recolector y de expulsar los tickets.
- Cuenta con dos lectores de código de barras, encargados de leer el código de los tickets de estacionamiento y de transmitir dicho dato hacia la tarjeta controladora del recolector de tickets por medio del protocolo RS232.
- Cuenta con un puerto RS232 para transmitir la información leída de los tickets y para recibir las órdenes de qué hacer con dichos tickets.
- Cuenta con un sensor de entrada y un sensor de posición de lectura, los cuales se encargan de detectar el ticket en las distintas posiciones del recolector.
- Cuenta con un botón de emergencias de expulsión del ticket, el cual se encarga de retirar el ticket en caso de que la lectora se encuentre esperando la respuesta desde la computadora.

FUNCIONES

- Al sensar el ticket de entrada el motor debe girar para colocar el ticket en la posición de lectura.
- Al detectar un ticket en posición de lectura debe detener el motor y el lector debe leer el código de barras.
- En todos los casos que se encienda el motor debe girar con un tiempo máximo de giro como sistema de seguridad.
- En el caso de que se inserte un ticket y por cualquier motivo no se pueda leer el código de barras, el recolector sacará parcialmente el ticket y volverá a meter dentro del recolector con la intención de realizar la lectura nuevamente. Si después del segundo intento no se consigue la lectura, el ticket será expulsado completamente del recolector.
- Expulsión automática del ticket. En el caso de que se lea un ticket y pase un tiempo X sin recibir una orden desde el puerto PC, el recolector automáticamente expulsará el ticket.
- Expulsión automática del ticket al encenderse el recolector.
- Función de reinicio de proceso de lectura si se retira manualmente el ticket, aunque se encuentre en “Espera Indefinida”.
- Configuración de dígitos a leer del código de barras mediante comando por rs232.
- Configuración de velocidad de giro del motor mediante un comando por rs232.
- Configuración de sentido de giro del motor en caso de que el hardware se haya colocado mal, mediante un comando por rs232.
- Configuración de posicionamiento del ticket en el área de lectura mediante un comando por rs232.
- Configuración de posicionamiento del ticket en la boquilla al momento de expulsarlo mediante un comando por rs232.
- Configuración de posicionamiento del ticket en la boquilla para relectura mediante un comando por rs232.
- Configuración del tiempo que el ticket permanece en el área de lectura mediante un comando por rs232.
- Configuración del tiempo de espera de respuesta de la PC mediante un comando por rs232.
- Configuración del tiempo máximo de giro del motor mediante un comando por rs232.
- Comando por rs232 para reestablecer el recolector a sus valores predeterminados.

Recolector de tickets

CARACTERÍSTICAS DEL CIRCUITO IMPRESO



Medidas físicas: 121.8 x 77.4 mm
 Voltaje de alimentación: 12 VCD
 Consumo máximo: 700 mA

RT-C2

Recolector de tickets

COMANDOS POR RS232

COMANDO: o

Comando utilizado para expulsar el ticket del recolector.

NOTA: Este comando solo se ejecuta si existe un ticket dentro del recolector, en caso contrario no realiza ninguna acción.

COMANDO: i

Comando utilizado para almacenar el ticket dentro del recolector.

NOTA: Este comando solo se ejecuta si existe un ticket dentro del recolector, en caso contrario no realiza ninguna acción.

COMANDO: p

Pausa la expulsión automática del ticket dentro por un tiempo indefinido. Las formas de cancelar la pausa indefinida son:

1. Recibiendo cualquiera de los siguientes comandos: i, o.
2. Retirando manualmente el ticket dentro del recolector.

NOTA: Este comando solo se ejecuta si existe un ticket dentro del recolector, en caso contrario no realiza ninguna acción.

COMANDO: s

Devuelve el valor de las constantes guardadas en la EEPROM del microcontrolador.

COMANDO: re

Establece los valores predeterminados de todas las variables utilizadas por el recolector.

COMANDO: rx:XX

Este comando establece la cantidad de dígitos que podrán ser recibidos del lector de código de barras.

Donde:

rx: es una cadena constante

XX representa la cantidad de dígitos que podrán recibirse y toma valores entre 00 y 50. Si se coloca el numero 00, el lector admitirá y enviará cualquier cadena, siempre y cuando la longitud de su cadena sea un número entre 01 y 50.

Ejemplos:

rx:00	Correcto:	configura la recepción de códigos de barras de cualquier longitud (predeterminado).
rx:10	Correcto:	configura la recepción de códigos de barras de 10 dígito.
rx:1	Incorrecto:	esta acción no genera ninguna respuesta en el recolector de tickets.
rx:52	Incorrecto:	este valor se encuentra fuera del rango.

RT-C2

Recolector de tickets

COMANDO: mv:XX

Este comando controla la velocidad con la que gira el motor de pasos.

Donde:

mv: es una cadena constante

XX representa el múltiplo del retardo entre los pasos del motor de pasos. Toma valores entre 01 y 99.

El paso mínimo es de 1.25ms. Para calcular el tiempo que tarda cada paso se debe multiplicar el paso mínimo por el valor de XX.

Ejemplos:

mv:01	Correcto:	configura el motor de pasos a su máxima velocidad con un retardo de paso de 1.25ms (predeterminado).
mv:50	Correcto:	configura el motor de pasos a la mitad de su velocidad con un retardo de paso de 62.5ms.
mv:99	Correcto:	configura el motor de pasos a su mínima velocidad con un retardo de paso de 123.75ms.
mv:00	Incorrecto:	este comando no genera ninguna respuesta en el recolector de tickets.
mv:1	Incorrecto:	esta acción no genera ninguna respuesta en el recolector de tickets.

ADVERTENCIA: Si se realiza el cambio de velocidad, medir el consumo de la placa, ya que entre más lento se mueve el motor, mayor consumo de corriente y se corre el riesgo de quemar el driver L298. Máximo consumo aceptado por el driver L298, 4 Amperes. Máximo aceptado sin presentar calentamiento 1.5 Amperes. Si se pretende usar a más de 1.5 Amperes, utilizar disipador de calor.

COMANDO: md:XX

Este comando controla el sentido de giro del motor. Sirve para invertir el giro del motor de pasos, en caso de que el motor gire en sentido inverso.

Donde:

md: es una cadena constante

XX controla el sentido de giro de motor de pasos. Toma valores entre 00 y 01. Sentido de giro normal 00 y sentido de giro invertido 01.

Ejemplos:

md:00	Correcto:	configura el sentido de giro normal del motor de pasos (predeterminado).
md:01	Correcto:	configura el sentido de giro invertido del motor de pasos.
md:05	Incorrecto:	este comando no genera ninguna respuesta en el recolector de tickets.
md:18	Incorrecto:	este comando no genera ninguna respuesta en el recolector de tickets.

COMANDO: mn:XXX

Este comando controla la posición del ticket en el área de lectura.

Donde:

mn: es una cadena constante

XXX determina la cantidad de pasos extra que dará el motor después de que el sensor de lectura se haya activado, esto permite ajustar la posición del ticket en el área de lectura. Toma valores entre 000 y 999.

Ejemplos:

mn:050	Correcto:	establece 50 pasos extra del motor para colocar el ticket en el área de lectura (predeterminado).
mn:100	Correcto:	establece 100 paso extra del motor para colocar el ticket en el área de lectura.
mn:0	Incorrecto:	este comando no genera ninguna respuesta en el recolector de tickets.
mn:01	Incorrecto:	esta acción no genera ninguna respuesta en el recolector de tickets.

RT-C2

Recolector de tickets

COMANDO: **mt:XXX**

Este comando controla la posición del ticket en una expulsión completa.

Donde:

mt: es una cadena constante

XXX determina la cantidad de pasos extra que dará el motor después de que el sensor de lectura se haya desactivado, esto permite ajustar la posición del ticket en la boquilla del recolector. Toma valores entre 000 y 999.

Ejemplos:

- | | | |
|--------|-------------|--|
| mt:0 | Incorrecto: | este comando no genera ninguna respuesta en el recolector de tickets. |
| mt:01 | Incorrecto: | esta acción no genera ninguna respuesta en el recolector de tickets. |
| mt:001 | Correcto: | establece 1 paso extra del motor para colocar el ticket en la boquilla del recolector. |
| mt:100 | Correcto: | establece 100 pasos extra del motor para colocar el ticket en la boquilla del recolector (predeterminado). |

COMANDO: **mc:XXX**

Este comando controla la posición del ticket en una expulsión parcial.

Donde:

mc: es una cadena constante

XXX determina la cantidad de pasos extra que dará el motor después de que el sensor de lectura se haya desactivado, esto permite ajustar la posición del ticket en la boquilla del recolector, sin llegar a sacarlo del recolector. Toma valores entre 000 y 999.

Ejemplos:

- | | | |
|--------|-------------|---|
| mc:0 | Incorrecto: | este comando no genera ninguna respuesta en el recolector de tickets. |
| mc:00 | Incorrecto: | esta acción no genera ninguna respuesta en el recolector de tickets. |
| mc:000 | Correcto: | establece 1 paso extra del motor para colocar el ticket en la boquilla del recolector (predeterminado). |
| mc:500 | Correcto: | establece 500 pasos extra del motor para colocar el ticket en la boquilla del recolector. |

COMANDO: **rl:XX**

Este comando establece el tiempo que el ticket espera en el área de lectura esperando la lectura de su código de barras.

Donde:

rl: es una cadena constante

XX representa el tiempo de espera en segundos y toma valores entre 00 y 10.

Ejemplos:

- | | | |
|-------|-------------|--|
| rl:01 | Correcto: | configura un tiempo de espera igual a 1 segundo (predeterminado). |
| rl:02 | Correcto: | configura un tiempo de espera igual a 2 segundos. |
| rl:1 | Incorrecto: | esta acción no genera ninguna respuesta en el recolector de tickets. |
| rl:11 | Incorrecto: | este valor se encuentra fuera del rango. |

COMANDO: **rc:XX**

Este comando establece el tiempo que el ticket espera en el área de lectura esperando una respuesta de la Computadora.

Donde:

rc: es una cadena constante

XX representa el tiempo de espera en segundos y toma valores entre 00 y 30.

Ejemplos:

- | | | |
|-------|-------------|--|
| rc:10 | Correcto: | configura un tiempo de espera igual a 10 segundo (predeterminado). |
| rc:20 | Correcto: | configura un tiempo de espera igual a 20 segundos. |
| rc:1 | Incorrecto: | esta acción no genera ninguna respuesta en el recolector de tickets. |

RT-C2

Recolector de tickets

rc:31 Incorrecto: este valor se encuentra fuera del rango.

RT-C2

Recolector de tickets

COMANDO: rt:XX

Este comando establece el tiempo máximo que por seguridad puede girar el motor.

Donde:

rt: es una cadena constante

XX representa el tiempo de espera en segundos y toma valores entre 01 y 10.

Ejemplos:

rt:02	Correcto:	configura un tiempo de espera igual a 2 segundo (predeterminado).
rt:04	Correcto:	configura un tiempo de espera igual a 4 segundos.
rt:1	Incorrecto:	esta acción no genera ninguna respuesta en el recolector de tickets.
rt:11	Incorrecto:	este valor se encuentra fuera del rango.

RESPUESTAS POR RS232

COMANDO: X

Este comando se envía si después de intentar realizar la lectura de un ticket no se obtiene el código.

COMANDO: N

Este comando se envía si el recolector está configurado para recibir una cantidad establecida de dígitos y recibe una cadena con más o menos números, envía una letra "N" como muestra de que se ha introducido un ticket incorrecto.

RT-C2

Recolector de tickets

ACERCA DE LAS ACTUALIZACIONES

RT-B5

- Se agregó función para controlar el posicionamiento del ticket con el comando *ba:XX*:

RT-B6

- Se cambiaron los pines de comunicación serial.
- Se eliminó botón para expulsar ticket y se eliminó conexión de pin pwm con el driver L298 para futuro control de velocidad.
- Se agregó un puerto DB-9 para conectar un 2do lector de código de barras.

RT-C1

General:

- Se cambió de motor, pasó de ser un motor de CD a un motor de pasos.

Hardware:

- Se adaptó el circuito del L298 para controlar el motor de pasos, el cual se alimenta directamente de la fuente de 12V después del puente diodos aproximadamente 10.4V.
- Se agregaron diodos 1N5404 para bajar el voltaje de la fuente y no sobrecalentar el 7805.
- Se distribuyó el consumo de corriente entre el 7805 y el LM317. El regulador LM317 se configuró para suministrar 5V, el cual únicamente alimenta a las lectoras de código de barras y todo lo demás lo suministra el 7805.
- El 7805, LM317 y TIP42 siguen generando calor, pero se cree que no es lo suficientemente alto para necesitar un disipador, por lo que se retiraron los disipadores del circuito.
- Se aumentó la resistencia del sensor de entrada para disminuir su consumo. Pasó de una resistencia de 100 Ohms a una de 330 Ohms, el consumo pasó de 35mA a 10mA.
- Se aumentó la resistencia del sensor de área de lectura para disminuir su consumo. Pasó de una resistencia de 100 Ohms a una de 1200hms, el consumo pasó de 35mA a una 30mA. Este sensor no debe bajarse más la corriente porque puede causar problemas para detectar tickets que entren mal.
- Se quitaron los transistores de los LEDs de los pines que comparten con el ICSP y se conectaron directamente, se probó que no causa ningún problema.
- Se colocó nuevamente el botón de expulsión de emergencia.

Errores de Hardware:

Los leds de los sensores QRD1114 están puestos a la salida del regulador de 5V, pasarlos a la salida del puente de diodo, de esta manera se reducirá el calentamiento en el regulador.

Software:

- Se adaptó el código para controlar el motor de pasos.
- Se agregó un comando por rs232 para posicionar el ticket en el área de lectura.
- Se agregó un comando por rs232 para posicionar el ticket en la boquilla.
- Se agregó un comando por rs232 para controlar el tiempo de encendido del motor.
- Se agregó un comando por rs232 para controlar el tiempo de espera en el área de lectura.
- Se agregó un comando por rs232 para controlar el tiempo de espera de respuesta de la PC.
- Se agregó un comando por rs232 para controlar la velocidad de giro del motor.
- Se agregó un comando por rs232 para invertir el sentido de giro del motor en caso de que este gire en sentido contrario.
- Se agregó un comando por rs232 para restablecer todas la variables a sus valores predeterminados.