RFID Card Reader, Serial (#28140)
RFID Card Reader, USB (#28340)

Introduction

Designed in cooperation with Grand Idea Studio (www.grandideastudio.com), the Parallax Radio Frequency Identification (RFID) Card Readers provide a low-cost solution to read passive RFID transponder tags up to 4 inches away. The RFID Card Readers can be used in a wide variety of hobbyist and commercial applications, including access control, user identification, robotics navigation, inventory tracking, payment systems, car immobilization, and manufacturing automation. The RFID Card Reader is available in two versions: A TTL-level serial interface for use with a microcontroller and a USB interface for direct connection to a computer.

Features

- Low-cost method for reading passive, 125 kHz RFID transponder tags
- Two easy-to-use versions: Serial interface for microcontrollers and USB for direct connection to PC, Macintosh, or Linux machines
- Bi-color LED for visual indication of status
RFID Compatibility

The Parallax RFID Card Reader works exclusively with the EM Microelectronics EM4100-family of passive read-only transponder tags. Each transponder tag contains a unique, read-only identifier (one of 2^60, or 1,099,511,627,776 possible combinations).

A variety of different tag types and styles exist, with the most popular made available from Parallax.

Connections (Serial)

The Parallax RFID Card Reader Serial version easily interfaces to any host microcontroller using only four connections (VCC, /ENABLE, SOUT, GND).

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin Name</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>P</td>
<td>System power. +5V DC input.</td>
</tr>
<tr>
<td>2</td>
<td>/ENABLE</td>
<td>I</td>
<td>Module enable pin. Active LOW digital input. Bring this pin LOW to enable the RFID reader and activate the antenna.</td>
</tr>
<tr>
<td>3</td>
<td>SOUT</td>
<td>O</td>
<td>Serial output to host. TTL-level interface, 2400 bps, 8 data bits, no parity, 1 stop bit.</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>G</td>
<td>System ground. Connect to power supply's ground (GND) terminal.</td>
</tr>
</tbody>
</table>

Note: Type: I = Input, O = Output, P = Power, G = Ground

Use the following example circuit for connecting the Parallax RFID Card Reader:
Connections (USB)

The Parallax RFID Card Reader USB version can be connected directly to any PC, Macintosh, or Linux machine that has a USB port and the appropriate drivers installed. The module is powered from the host computer’s USB port and uses an industry-standard FTDI FT232R device to provide the USB connectivity. FTDI drivers are available from www.ftdichip.com/Drivers/VCP.htm.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Port Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX</td>
<td>Serial Receive</td>
<td>Serial output to host. 2400 bps, 8 data bits, no parity, 1 stop bit.</td>
</tr>
<tr>
<td>DTR</td>
<td>Data Terminal Ready</td>
<td>Module enable. Bring the DTR line HIGH to enable the RFID reader and activate the antenna. Bring the DTR line LOW to disable the RFID reader.</td>
</tr>
</tbody>
</table>

When the Parallax RFID Card Reader is connected to the host computer, it will appear as a Virtual COM port and will have a COM port number automatically assigned to it. This COM port can be accessed by any software application, programming language, or interface that provides COM port connectivity and will allow you to read the data stream transmitted by the module.

An example program, including Visual Basic/VB.net source code, for reading tags in Windows XP/Vista is available from the 28340 product page at www.parallax.com.

The Debug Terminal within the Parallax BASIC Stamp Editor (www.parallax.com/basicstampsoftware) provides functionality to set the state of a COM port’s DTR line. Checking the DTR box in the toolbar will activate the RFID Card Reader.
Usage
A visual indication of the state of the RFID Card Reader is given with the on-board LED. When the module is successfully powered-up and is in an idle state, the LED will be GREEN. When the module is in an active state searching for or communicating with a valid tag, the LED will be RED.

The RFID Card Reader Serial version is activated via the /ENABLE pin on the module’s 4-pin header. When the RFID Card Reader is powered and /ENABLE is pulled LOW, the module will enter the active state. When /ENABLE is pulled HIGH or left unconnected, the module will enter the idle state.

The RFID Card Reader USB version is activated via the DTR line of the USB Virtual COM port. When the DTR line is set HIGH, the module will enter the active state. When the DTR line is set LOW, the module will enter the idle state.

The face of the RFID tag should be held parallel to the front or back face of the antenna (where the majority of RF energy is emitted). If the tag is held sideways (for example, perpendicular to the antenna), you’ll either get no reading or a poor reading distance. Only one transponder tag should be held up to the antenna at any time. The use of multiple tags at one time will cause tag collisions and the reader may not detect any of them. The tags available in the Parallax store have a read distance of approximately 4 inches. Actual distance may vary slightly depending on the size of the transponder tag and environmental conditions of the application.

Communication Protocol
All communication is 8 data bits, no parity, 1 stop bit, and least significant bit first (8N1) at 2400 bps.

The RFID Card Reader Serial version transmits data as 5 V TTL-level, non-inverted asynchronous serial.

The RFID Card Reader USB version transmits the data through the USB Virtual COM Port driver. This allows easy access to the serial data stream from any software application, programming language, or interface that can communicate with a COM port.

When the RFID Card Reader is active and a valid RFID transponder tag is placed within range of the activated reader, the tag’s unique ID will be transmitted as a 12-byte printable ASCII string serially to the host in the following format:

```
Error! Objects cannot be created from editing field codes.
```

The start byte and stop byte are used to easily identify that a correct string has been received from the reader (they correspond to line feed and carriage return characters, respectively). The middle ten bytes are the actual tag’s unique ID. For example, for a tag with a valid ID of 0F0184F07A, the following bytes would be sent: 0x0A, 0x30, 0x46, 0x30, 0x31, 0x38, 0x34, 0x46, 0x30, 0x37, 0x41, 0x0D.

Interference
The Parallax RFID Card Reader, like many RF devices, may experience RF noise in its frequency range. This may cause the reader to transmit a spurious tag response when no tag is near the unit. This will not affect most uses of the RFID Card Reader. To avoid treating spurious responses as legitimate tags, it is recommended to read two responses in a row within a given amount of time (for example, one second) to ensure that you are reading a valid tag and not a “tag” generated by noise.

DC Characteristics
At $V_{CC} = +5.0V$ and $T_A = 25^\circ C$ unless otherwise noted.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Test Conditions</th>
<th>Specification</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage</td>
<td>$V_{CC}$</td>
<td>---</td>
<td>Min. 4.5</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 5.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max. 5.5</td>
<td></td>
</tr>
<tr>
<td>Supply Current, Idle</td>
<td>$I_{IDLE}$</td>
<td>---</td>
<td>---</td>
<td>mA</td>
</tr>
<tr>
<td>Supply Current, Active</td>
<td>$I_{CC}$</td>
<td>---</td>
<td>Min. 10</td>
<td>mA</td>
</tr>
<tr>
<td>Input LOW voltage</td>
<td>$V_{IL}$</td>
<td>+4.5V &lt;= $V_{CC}$ &lt;= +5.5V</td>
<td>---</td>
<td>V</td>
</tr>
<tr>
<td>Input HIGH voltage</td>
<td>$V_{IH}$</td>
<td>+4.5V &lt;= $V_{CC}$ &lt;= +5.5V</td>
<td>Min. 0.8</td>
<td>V</td>
</tr>
<tr>
<td>Output LOW voltage</td>
<td>$V_{OL}$</td>
<td>$V_{CC} = +4.5V$</td>
<td>---</td>
<td>V</td>
</tr>
<tr>
<td>Output HIGH voltage</td>
<td>$V_{OH}$</td>
<td>$V_{CC} = +4.5V$</td>
<td>Min. 0.6</td>
<td>V</td>
</tr>
</tbody>
</table>

**Absolute Maximum Ratings**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Temperature</td>
<td>-40ºC to +85ºC</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>-55ºC to +125ºC</td>
</tr>
<tr>
<td>Supply Voltage ($V_{cc}$)</td>
<td>+4.5V to +5.5V</td>
</tr>
<tr>
<td>Ground Voltage ($V_{ss}$)</td>
<td>0V</td>
</tr>
<tr>
<td>Voltage on any pin with respect to $V_{ss}$</td>
<td>-0.3V to +7.0V</td>
</tr>
</tbody>
</table>

**NOTICE:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**RFID Technology Overview**

*Material in this section is based on information provided by the RFID Journal ([www.rfidjournal.com](http://www.rfidjournal.com)).*

Radio Frequency Identification (RFID) is a generic term for non-contacting technologies that use radio waves to automatically identify people or objects. There are several methods of identification, but the most common is to store a unique serial number that identifies a person or object on a microchip that is attached to an antenna. The combined antenna and microchip are called an "RFID transponder" or "RFID tag" and work in combination with an "RFID reader" (sometimes called an "RFID interrogator").

An RFID system consists of a reader and one or more tags. The reader's antenna is used to transmit radio frequency (RF) energy. Depending on the tag type, the energy is "harvested" by the tag's antenna and used to power up the internal circuitry of the tag. The tag will then modulate the electromagnetic waves generated by the reader in order to transmit its data back to the reader. The reader receives the modulated waves and converts them into digital data.

There are two major types of tag technologies. "Passive tags" are tags that do not contain their own power source or transmitter. When radio waves from the reader reach the chip's antenna, the energy is converted by the antenna into electricity that can power up the microchip in the tag (typically via inductive coupling). The tag is then able to send back any information stored on the tag by modulating the reader's electromagnetic waves. "Active tags" have their own power source and transmitter. The power source, usually a battery, is used to run the microchip's circuitry and to broadcast a signal to a reader. Due to the fact that passive tags do not have their own transmitter and must reflect their signal to the reader, the reading distance is much shorter than with active tags. However, active tags are typically larger, more expensive, and require occasional service.
Frequency refers to the size of the radio waves used to communicate between the RFID system components. Just as you tune your radio to different frequencies in order to hear different radio stations, RFID tags and readers must be tuned to the same frequency in order to communicate effectively. RFID systems typically use one of the following frequency ranges: low frequency (or LF, around 125 kHz), high frequency (or HF, around 13.56 MHz), ultra-high frequency (or UHF, around 868 and 928 MHz), or microwave (around 2.45 and 5.8 GHz).

The read range of a tag ultimately depends on many factors: the frequency of RFID system operation, the power of the reader, and interference from other RF devices. Balancing a number of engineering trade-offs (antenna size v. reading distance v. power v. manufacturing cost), the Parallax RFID Card Reader’s antenna was designed specifically for use with low-frequency (125 kHz) passive tags with a read distance of around 4 inches.

### BASIC Stamp® 1 Program

The following code examples read tags from a RFID Card Reader and compare the values to known tags (stored in an EEPROM table).

```basic
' -----[ Program Description ]--------------------------------------------------------
' Reads tags from a Parallax RFID reader and compares to known tags (stored in EEPROM table). If tag is found, the program will disable a lock.

' -----[ Revision History ]------------------------------------------------------------

' -----[ I/O Definitions ]-------------------------------------------------------------

SYMBOL Enable = 0                     ' low = reader on
SYMBOL RX = 1                         ' serial from reader
SYMBOL Spkr = 2                       ' speaker output
SYMBOL Latch = 3                      ' lock/latch control

' -----[ Constants ]------------------------------------------------------------------

SYMBOL LastTag = 2                    ' 3 tags; 0 to 2

' -----[ Variables ]------------------------------------------------------------------

SYMBOL tag0 = B0                      ' RFID bytes buffer
SYMBOL tag1 = B1
SYMBOL tag2 = B2
```
SYMBOL tag3 = B3
SYMBOL tag4 = B4
SYMBOL tag5 = B5
SYMBOL tag6 = B6
SYMBOL tag7 = B7
SYMBOL tag8 = B8
SYMBOL tag9 = B9

SYMBOL tagNum = B10          ' from EEPROM table
SYMBOL pntr = B11            ' pointer to char in table
SYMBOL char = B12            ' character from table

' ----- [ EEPROM Data ]-----------------------------------------------------

Tags:
EEPROM ("0F0184F20B")                   ' valid tags
EEPROM ("0F01D9D263")
EEPROM ("04129C1B43")
EEPROM ("000000000000")                ' space for other tags
EEPROM ("000000000000")

' ----- [ Initialization ]--------------------------------------------------

Reset:
HIGH Enable                                   ' turn off RFID reader
LOW Latch                                     ' lock the door!

' ----- [ Program Code ]----------------------------------------------------

Main:
LOW Enable                                    ' activate the reader
SERIN RX, T2400, ($0A)                        ' wait for header
SERIN RX, T2400, tag0, tag1, tag2, tag3, tag4 ' get tag bytes
SERIN RX, T2400, tag5, tag6, tag7, tag8, tag9
HIGH Enable                                   ' deactivate reader

Check_List:
FOR tagNum = 0 TO LastTag                     ' scan through known tags
  pntr = tagNum * 10 + 0 : READ pntr, char    ' read char from DB
  IF char <> tag0 THEN Bad_Char
  IF char <> tag1 THEN Bad_Char
  IF char <> tag2 THEN Bad_Char
  IF char <> tag3 THEN Bad_Char
  IF char <> tag4 THEN Bad_Char
  IF char <> tag5 THEN Bad_Char
  IF char <> tag6 THEN Bad_Char
  IF char <> tag7 THEN Bad_Char
  IF char <> tag8 THEN Bad_Char
  IF char <> tag9 THEN Bad_Char
  GOTO Tag_Found                              ' all match -- good tag

Bad_Char:
**BASIC Stamp® 2 Program**

The following code examples read tags from a RFID Card Reader and compare the values to known tags (stored in an EEPROM table).

```
NEXT

Bad_Tag:
    SOUND Spkr, (25, 80)          ' groan
    PAUSE 1000
    GOTO Main

Tag_Found:
    DEBUG #tagNum, CR            ' for testing
    HIGH Latch                   ' remove latch
    SOUND Spkr, (114, 165)       ' beep
    LOW Latch                    ' restore latch
    GOTO Main

END
```

---

### Program Description

Reads tags from a Parallax RFID reader and compares to known tags (stored in EEPROM table). If tag is found, the program will disable a lock.

---

### Revision History

---

### I/O Definitions

<table>
<thead>
<tr>
<th>PIN</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>low = reader on</td>
</tr>
<tr>
<td>1</td>
<td>serial from reader</td>
</tr>
<tr>
<td>2</td>
<td>speaker output</td>
</tr>
<tr>
<td>3</td>
<td>lock/latch control</td>
</tr>
</tbody>
</table>

---

### Constants

```
#SELECT $STAMP
#CASE BS2, BS2E, BS2PE
    T1200 CON 813
    T2400 CON 396
    T4800 CON 188
    T9600 CON 84
    T19K2 CON 32
```

---
# CASE BS2SX, BS2P
  T1200 CON 2063
  T2400 CON 1021
  T4800 CON 500
  T9600 CON 240
  T19K2 CON 110
  TMidi CON 60
  T38K4 CON 45
# CASE BS2PX
  T1200 CON 3313
  T2400 CON 1646
  T4800 CON 813
  T9600 CON 396
  T19K2 CON 188
  TMidi CON 108
  T38K4 CON 84
# ENDSELECT

SevenBit CON $2000
Inverted CON $4000
Open CON $8000
Baud CON T2400

# SELECT $STAMP
  # CASE BS2, BS2E
  TmAdj CON $100 ' x 1.0 (time adjust)
  FrAdj CON $100 ' x 1.0 (freq adjust)
  # CASE BS2SX
  TmAdj CON $280 ' x 2.5
  FrAdj CON $066 ' x 0.4
  # CASE BS2P
  TmAdj CON $3C5 ' x 3.77
  FrAdj CON $044 ' x 0.265
  # CASE BS2PE
  TmAdj CON $100 ' x 1.0
  FrAdj CON $0AA ' x 0.665
  # CASE BS2Px
  TmAdj CON $607 ' x 6.03
  FrAdj CON $2A ' x 0.166
# ENDSELECT

LastTag CON 3

# DEFINE __No_SPRAM = ($STAMP < BS2P) ' does module have SPRAM?

' -----[ Variables ]--------------------------------------------

# IF __No_SPRAM #THEN
  buf VAR Byte(10) ' RFID bytes buffer
# ELSE
  chkChar VAR Byte ' character to test
# ENDIF

  tagNum VAR Nib ' from EEPROM table
  idx VAR Byte ' tag byte index
  char VAR Byte ' character from table

Copyright © Parallax Inc. RFID Card Reader Serial & USB (#28140 / 28340) v2.3 10/9/2014 Page 9 of 11
' ------[ EEPROM Data ]-----------------------------------------------------
Tag1   DATA   "0F0184F20B"          ' valid tags
Tag2   DATA   "0F01D9D263"
Tag3   DATA   "04129C1B43"
Name0  DATA   "Unauthorized", CR, 0
Name1  DATA   "George Johnston", CR, 0
Name2  DATA   "Dick Miller", CR, 0
Name3  DATA   "Mary Evans", CR, 0

' ------[ Initialization ]---------------------------------------------------
Reset:
  HIGH Enable                               ' turn on RFID reader
  LOW Latch                                 ' lock the door!

' ------[ Program Code ]-----------------------------------------------------
Main:
  LOW Enable                               ' activate the reader
  #IF __No_SPRAM #THEN
    SERIN RX, T2400, [WAIT($0A), STR buf\10]    ' wait for hdr + ID
  #ELSE
    SERIN RX, T2400, [WAIT($0A), SPSTR 10]
  #ENDIF
  HIGH Enable                               ' deactivate reader

Check_List:
  FOR tagNum = 1 TO LastTag                     ' scan through known tags
    FOR idx = 0 TO 9                            ' scan bytes in tag
      READ (tagNum - 1 * 10 + idx), char        ' get tag data from table
      #IF __No_SPRAM #THEN
        IF (char <> buf(idx)) THEN Bad_Char     ' compare tag to table
      #ELSE
        GET idx, chkChar                        ' read char from SPRAM
        IF (char <> chkChar) THEN Bad_Char      ' compare to table
      #ENDIF
    NEXT
  NEXT
  GOTO Tag_Found                              ' all bytes match!

Bad_Char:
    NEXT

Bad_Tag:
    tagNum = 0
    GOSUB Show_Name                           ' print message
    FREQUOT Spkr, 1000 */ TmAdj, 115 */ FrAdj ' groan
    PAUSE 1000
    GOTO Main

Tag_Found:
    GOSUB Show_Name                           ' print name
    HIGH Latch                                ' remove latch
    FREQUOT Spkr, 2000 */ TmAdj, 880 */ FrAdj ' beep
    LOW Latch                                 ' restore latch
    GOTO Main

END
Show Name:
    DEBUG DEC tagNum, ": 
    LOOKUP tagNum, 
        [Name0, Name1, Name2, Name3], idx      ' point to first character
    DO
        READ idx, char                              ' read character from name
        IF (char = 0) THEN EXIT                     ' if 0, we're done
        DEBUG char                                  ' otherwise print it
        idx = idx + 1                               ' point to next character
    LOOP
    RETURN