

## RFID Door Lock



Version 1.0

October 2009

Cytron Technologies Sdn. Bhd.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

### OVERVIEW

This document describes the development of Cytron Technologies DIY (Do It Yourself) Project PR25. This project revises PR8: Read and Display and PR9: Password Door Security. However, this project improves the previous PR, where it can learn new RFID ID tag and read ID tag in two different communication ways, Wiegand and UART. Connection for UART and Wiegand RFID tag reader are provided on PCB to be extended. Schematic and source code is also provided.

### FEATURES

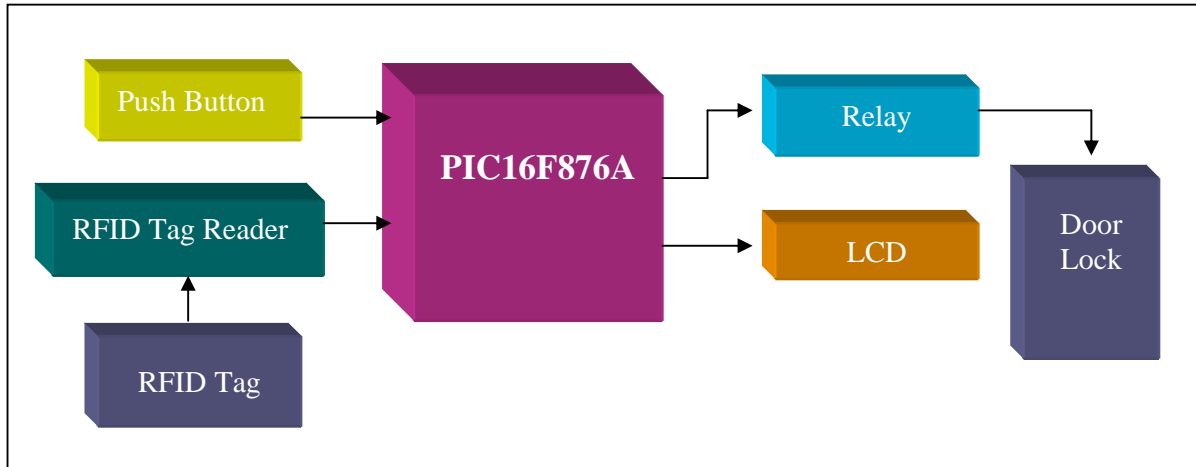
#### PIC16F876A

- 8-bit microcontroller with 22 I/O
- Operate with 5V supply
- Operating speed 20MHz

#### WIEGAND RFID TAG READER

- RFID tag reader with specification reader-to-card and binary reader-to-controller interface
- Standard 26-bit binary card data format
- Operate with 5V supply

## SYSTEM OVERVIEW



## GENERAL DESCRIPTION

PR25 is an open source microcontroller Do It Yourself kit. This PIC microcontroller based project is designed to develop a low cost RFID door lock. It combine the features and functions of PR8 (RFID: Read and Display) and PR9 (Password Door Security). When RFID tag is place near to RDIF reader, the reader can read RFID tag, further send the tag ID to the PIC microcontroller. After PIC microcontroller process the data, the tag ID will display on LCD in decimal number and only particular RFID tag can pass the door lock. However, this project improves the previous PR, where it can read ID tag in two different communication ways, Wiegand and UART. And connection for UART and Wiegand RFID tag reader are provided on PCB to be extended. Besides, it also can “learn” new ID tag. The new RFID tag saved can open the door lock for next time used. This DIY project also provides LCD (2x16 characters) for user to display the status. Buzzer is not provided as there is an internal build buzzer inside the RFID tag reader.

### PIC16F876A

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40- or 44-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. Features of the device:

- 256 bytes of EEPROM data memory
- self programming
- 2 Comparators
- 8 channels of 10-bit Analog-to-Digital (A/D) converter

- 2 capture/compare/PWM functions
- synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus
- Universal Asynchronous Receiver Transmitter (UART).

All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

#### 28-Pin PDIP, SOIC, SSOP

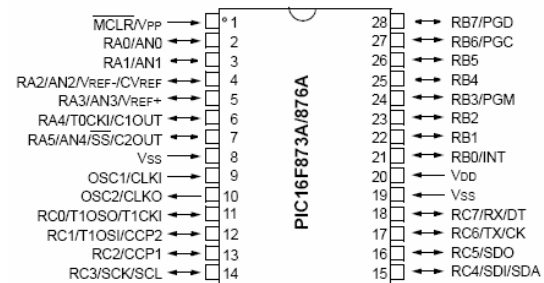


Figure 1

Figure 1 shows the pin diagram for PIC16F876A. For more information about the PIC microcontroller, please refer to the datasheet. The datasheet can be found in microchip web site at: <http://www.microchip.com>

## RFID Reader (Wiegand™ Format)

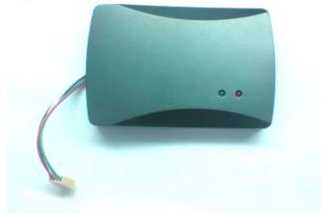


Figure 2

The term Wiegand is applied to several characteristics related to access control readers and cards. Unfortunately, the word is used carelessly and can lead to unnecessary confusion. Here are the basics. Wiegand is:

1. A specific reader-to-card interface
2. A specific binary reader-to-controller interface
3. An electronic signal carrying data
4. The standard 26-bit binary card data format
5. An electromagnetic effect
6. A card technology

For more information on how RFID work, please refer to Cytron's DIY project PR8 at [www.cytron.co.my](http://www.cytron.co.my)

## Power supply for the circuit

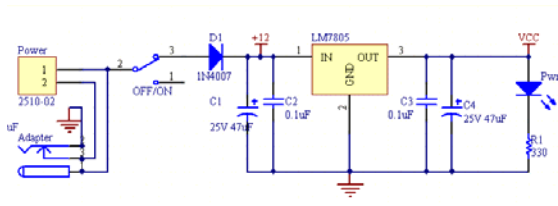


Figure 3

User can choose either AC to DC adaptor (not included in the DIY project set) or 9V-12V battery (not included in the DIY project set) to power up the circuit. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 12V. Anyhow, LM7805 will still generate some heat at 12V. There are two type of power connector for the circuit, DC plug (J1) and 2510-02 (Power Connector). Normally AC to DC adaptor can be plugged to J1 type connector.

Refer to Figure 3, the D1 is use to protect the circuit from wrong polarity supply. C1 and C2 is use to stabilize the voltage at the input side of the LM7805 voltage regulator, while the C3 and C4 is use to stabilize the voltage at the output side of the LM7805 voltage supply. LED is a green LED to indicate the power status of the circuit. R1 is resistor to protect LED from over current that will burn the LED.

## Push Button as input for PIC microcontroller

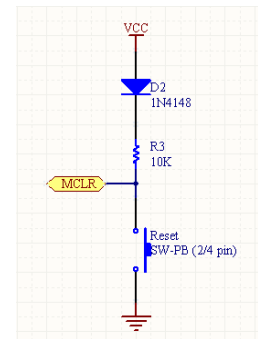


Figure 4

One I/O pin is needed for one push button as input of PIC microcontroller. The connection of the push button to the I/O pin is shown in Figure 3. The I/O pin should be pull up to 5V using a resistor (with value range 1K-10K) and this configuration will result an active-low input. When the button is being pressed, reading of I/O pin will be in logic 0, while when the button is not pressed, reading of that I/O pin will be logic 1.

## ICSP for programming PIC microcontroller

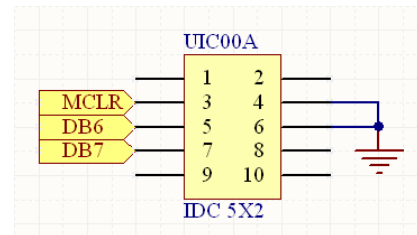


Figure 5

MCLR, RB6 and RB7 need to be connected to the USB In Circuit Programmer (UIC00A) to program the PIC microcontroller. The programmer (UIC00A) is not included in DIY project set since it can be used several time for different project set. User can also choose other type of PIC programmer to load the program.

For the instruction of using PIC programmer, please refer to the particular PIC programmer user's manual at cytron website.

## Relay

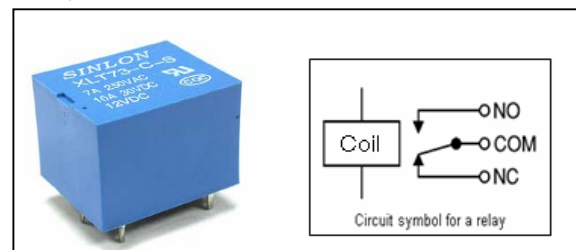


Figure 6

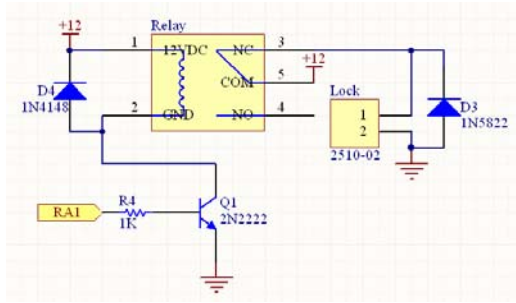


Figure 7

A relay (Figure6) is a simple electromechanical switch made up of an electromagnet and a set of contacts. Current flow through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be ON or OFF so relay have two switch positions and they are double throw (changeover) switches. Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits; the link is magnetic and mechanical. The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a *transistor* is usually used to amplify the small IC current to the larger value required for the relay coil. Relays are usually Single Pole Double Throw (SPDT) or Double Pole Double Throw (DPDT) but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available.

### Transistor

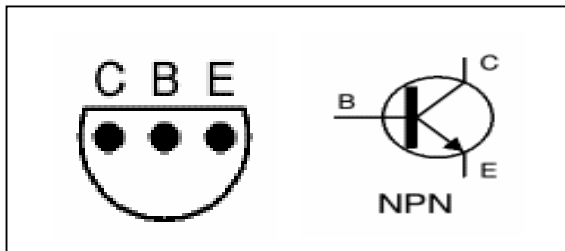


Figure 8

The transistor used to amplify the current for the relay is an NPN transistor, 2N2222A. The pin sequence is as shown in Figure 8. Beware that wrong sequence will burn the transistor.

### LED as output for PIC microcontroller

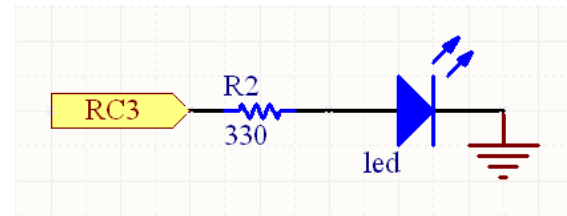


Figure 9

One I/O pin is designated for a LED as output of PIC microcontroller. The connection for a LED to I/O pin is shown in Figure9. The function of R2 is to protect the LED from over current that will burn the LED. When the output is in logic 1, the LED will ON, while when the output is in logic 0, the LED will OFF.

### Magnetic lock



Figure 10

A **magnetic lock** is a simple locking device that consists of an electromagnet and armature plate. By attaching the electromagnet to the door frame and the armature plate to the door, a current passing through the electromagnet attracts the armature plate holding the door shut. Unlike an electric strike a magnetic lock has no interconnecting parts and is therefore not suitable for high security applications because it is possible to bypass the lock by disrupting the power supply. Nevertheless, the strength of today's magnetic locks compare well with conventional door locks and cost less than conventional light bulbs to operate.

### HARDWARE

This project will require following hardware:

- 1 x PIC16F876A
- 1 x PR25 Printed Circuit Board (PCB)
- 1 x LCD (2x16 character)
- 1 x 12V relay
- 1 x RFID tag reader
- 1 x 9V-12V power supply
- Related electronic components

Please refer to Appendix A for the board layout of PR25. The board layout is provided free therefore Cytron Technologies will not be responsible for any further modification or improvement.

## Interface LCD (2x16 Character) with PIC16F876A

To use the LCD display, users have to solder 16 pin header pin to the LCD display. LCD used in this project is RT1602C, for other type of LCD, please refer to its data sheet.

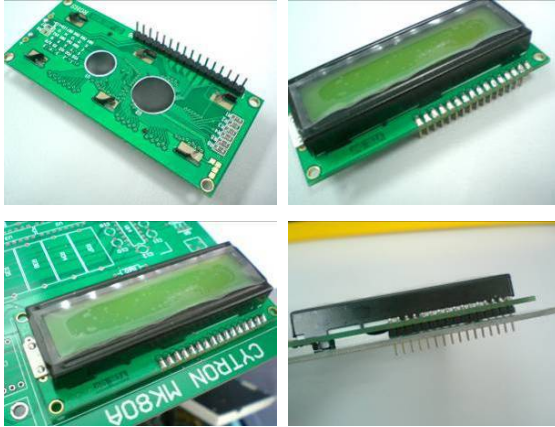


Figure 11

Figure 11 is a 2X16 character LCD. LCD connection pin and function of each pin is shown:

| Pin | Name | Pin function   | Connection                                      |
|-----|------|--|---|
| 1   | VSS  | Ground   | GND   |
| 2   | VCC  | Positive supply for LCD                              | 5V  |
| 3   | VEE  | Brightness adjust                                    | Connected to a trimmer to adjust LCD brightness |
| 4   | RS   | Select register, select instruction or data register | RA0   |
| 5   | R/W  | Select read or write                                 | GND   |
| 6   | E    | Start data read or write                             | RC5   |
| 7   | DB0  | Data bus pin   | RB0   |
| 8   | DB1  | Data bus pin   | RB1   |
| 9   | DB2  | Data bus pin   | RB2   |
| 10  | DB3  | Data bus pin   | RB3   |
| 11  | DB4  | Data bus pin   | RB4   |
| 12  | DB5  | Data bus pin   | RB5   |
| 13  | DB6  | Data bus pin   | RB6   |
| 14  | DB7  | Data bus pin   | RB7   |
| 15  | LED+ | Backlight positive input                             | 5V  |
| 16  | LED- | Backlight negative input                             | RC4   |

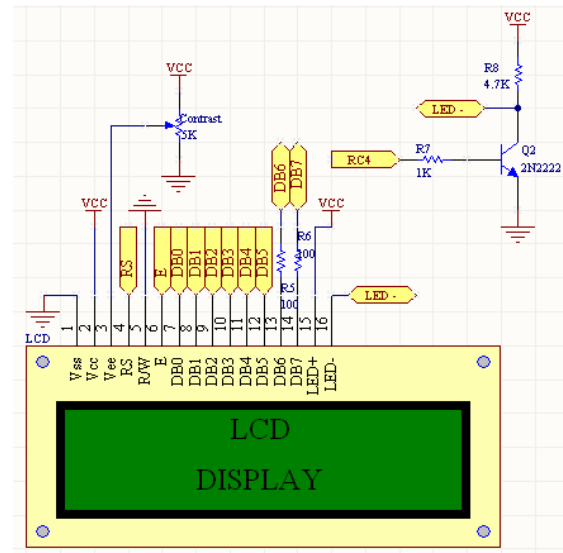


Figure 12

The transistor (Q2 2N2222) is use to provide more current for the LCD Backlight.

## Interface RFID tag with WIEGAND RFID tag reader

### 26-Bit Wiegand Format

The composition of the industry standard 26 Bit Wiegand format contains 8 bits for the facility code field and 16 bits for the ID number field. Mathematically these 8 facility code bits allow for a total of just 256 (0 to 255) facility codes, while the 16 ID number bits allow for a total of only 65,536 (0 to 65,535) individual ID's within each facility code. Due to the mathematical limitations of the 26-bit Wiegand format, code duplication might occur. Table 1 provides a summary the 26-bit Wiegand format.

| Bit Number    | Purpose                                |
|---------------|--|
| Bit 1         | Even parity over bits 2 to 13          |
| Bits 2 to 9   | Facility code (0 to 255); Bit 2 is MSB |
| Bits 10 to 25 | ID Number (0 to 65,535); Bit 10 is MSB |
| Bit 26        | Odd parity over bits 14 to 25          |

Table 1: 26-bit Wiegand Format

### Data Transmission in Wiegand 26 Standard

Figure 13 displays the timing pattern for data bits sent by the reader to the access control panel. This timing pattern falls within the Wiegand guidelines as proscribed by the SIA's Access Control Standard Protocol for the 26-Bit Wiegand Reader Interface (a Pulse Width time between 20 uS and 100 uS, and a Pulse Interval time between 200 uS and 20 mS).

The Data 1 and Data 0 signals are held at a logic high level (above the Voh level) until the reader is ready to send a data stream. The reader places data as

asynchronous low-going pulses (below the Vol level) on the Data 1 or Data 0 lines to transmit the data stream to the access control panel (the "saw-teeth" in Figure 13). The Data 1 and Data 0 pulses will not overlap or occur simultaneously. Table 2 provides the minimum and maximum allowable pulse width times (the duration of a pulse) and pulse interval times (the time between pulses) for Pyramid Series Readers.

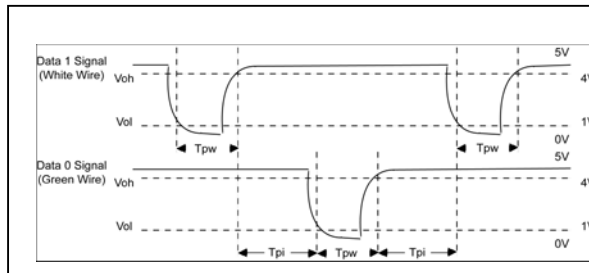


Figure13: Data Bit Timing Pattern

| Symbol | Description         | Pyramid Series Reader Typical Time |
|--------|---------------------|------------------------------------|
| Tpw    | Pulse Width Time    | 100 $\mu$ s                        |
| Tpi    | Pulse Interval Time | 1 ms                               |

Table 2: Pulse Times

As explained above, the Data 1 and Data 0 signals in RFID reader are held at a logic high level until there is a RFID tag is placed near to the RFID reader. When a RFID tag is scanned, the RFID will receive the data stream from RIFD tag.



Figure 14

Since there is 26 bit of data in a RFID tag, so once the RFID tag is scanned, the data will receive one bit of data start from data [0] to data [25] in the program. Each data is saved in the RFID tag reader with only bit 0 else equal to 0, which is 0...0000 000X.

For an example, if the tag ID is as below,

Bit 25 → 0  
1 0110 1001 1000 0100 0110 1010 1

The illustration of data process will be like this:

Data [0] = 0...0000 0001  
Data [1] = 0...0000 0000  
Data [2] = 0...0000 0001  
Data [3] = 0...0000 0000  
Data [4] = 0...0000 0001  
Data [5] = 0...0000 0000

Data [6] = 0...0000 0001  
Data [7] = 0...0000 0001  
Data [8] = 0...0000 0000  
⋮  
Data [25] = 0...0000 00001

In the program, we shift the current data and combine with previous saved data and repeat the loop until the last data. Using the example above, Data [0] will be saved in the variable convert1, and then Data [1] is shift to left by one bit to combine with the Data [0]. Variable convert1 is updated become 0...0000 0001 now and the loop keep going until last data.

```
for(i=0;i<8;i++) //loop for d
{
    convert1=(convert1<<1)|data[i+1]; //shift prev
}
for(i=0;i<16;i++) //loop for d
{
    convert2=(convert2<<1)|data[i+9]; //shift prev
}

id1[1]=convert1; //save converte
id2[1]=convert2; //save converte

database=2; //set database
lcd_clr(); //clear lcd
```

Figure 15

Note that data are store in two variable, convert1 and convert2. This is because Wiegand 26 bit is a special data format, bits 2 to 9 and bits 10 to 25 are grouped in two and they are use for different purpose (Facility Code and ID Number), as shown in Table 1 above. The special data format of Wiegand unable the two grouped data store in a variable, so id number array use id1 and id2 to save the data from convert1 and convert2.

## Interface WIEGAND RFID tag reader with PIC16F876A

For this project, user has to connect the wire of RFID reader to a 2510-04 female connector. There are 6 output wire of RFID reader, only 4 are used in this project. Be careful when plug in the 4 wires to the connector, connect them according to the table shown below.

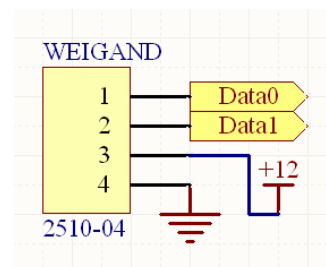


Figure 16

| Colour | Pin function | Connection |
|--------|--------------|------------|
| Red    | 12V          | 12V        |
| Black  | Ground       | GND        |
| Green  | Data 0       | RC1        |
| White  | Data1        | RC2        |
| Yellow | Led          | Not Used*  |
| Brown  | Buzzer       | Not Used*  |

\* = not connected

Table 3: Wire description

Connect only **four (green, white, red and black)** of the wire to 2510-04 female connector according to the colour of the wire. For more information about how to connect the wire to 2510-04 connector, please refer to getting start section.

### PCB circuit board

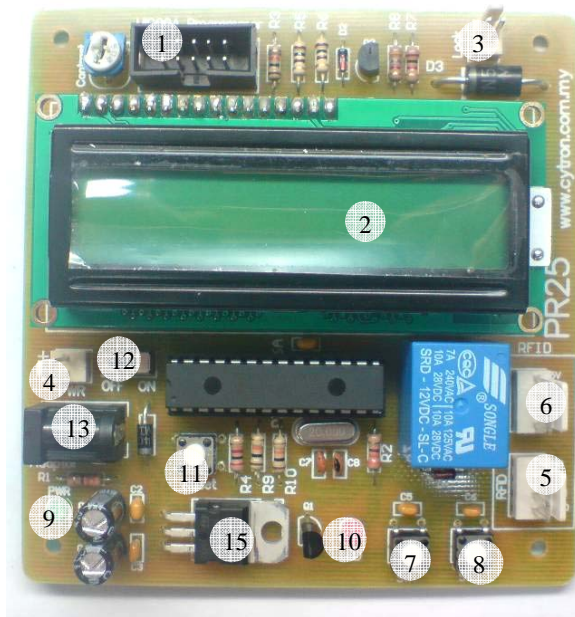


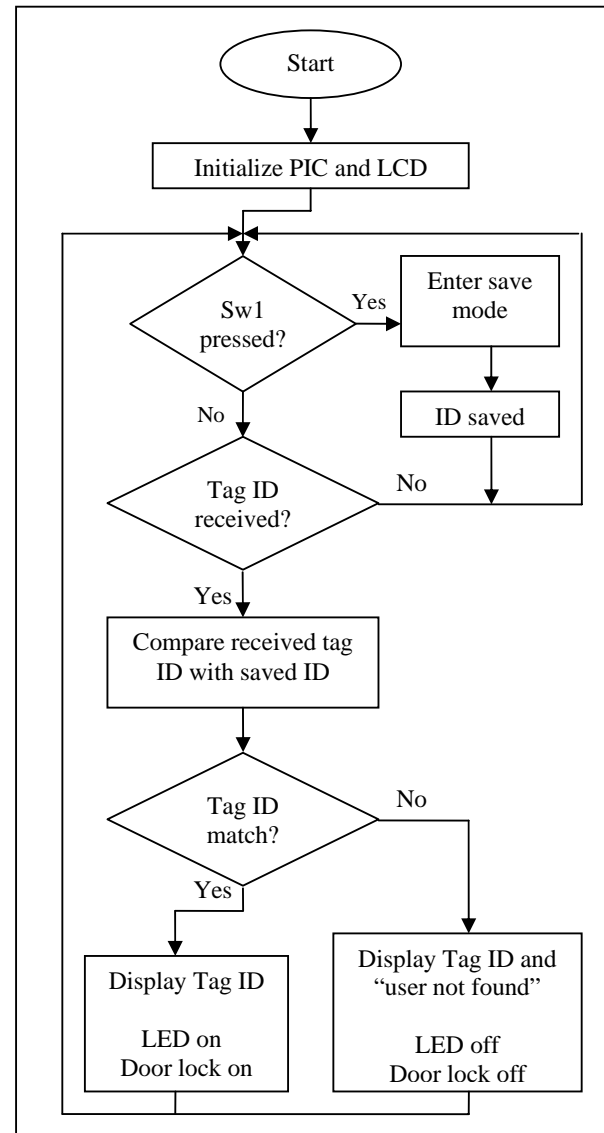
Figure 17 19

### Component:

1. Box header (To ICSP programmer).
2. Parallel LCD 2x16
3. 2510-02 connector for door lock.
4. 2510-02 connector for 9V battery or 12V battery to power up the circuit.
5. 2510-04 connector (RFID tag reader connector)
6. 2510-04 Connector (UART Connector)
7. SW1 push button to save mood.
8. SW2. This push button not used in sample source code provided.
9. LED (indicate power status of circuit)
10. LED (indicate the status of door lock)
11. Reset button (to reset the microcontroller)
12. Slide switch (to ON or OFF the circuit)
13. AC-DC adaptor socket (to use power supply from AC-DC adaptor)

### SOFTWARE

#### Flow Chart:



For more information about the software for this system, please refer to the source code provided. The explanation of each instruction is provided in the source code as the comment of each line.

The source code is provided free and Cytron Technologies will not be responsible for any further modification or improvement.

## GETTING START

User can obtain the hardware set for this project (PR25) either by online purchasing ([www.cytron.com.my](http://www.cytron.com.my)) or purchase it in Cytron Technologies Shop.

1. Once user has the hardware set, soldering process can be started. Please solder the electronic components one by one according the symbols or overlays on the Printed Circuit Board (PCB). Ensure the component value and polarity is correctly soldered. Please refer to PCB Layout in Appendix A.

**Caution:** Make sure all the connectors (2510) are soldered in proper side. Those electronic components have polarity such as capacitor, diode, PIC, LM7805 and LED should be soldered in right polarity or it may cause the circuit board fail to work.

**Warning:** Before the battery (Power) is plugged in, make sure the polarity is correct to prevent the explosion. Wrong polarity of capacitor also may cause explosion.

2. Connect the RFID reader to 2510-04 connector.
3. Connect the door lock to 2510-02 connector.

### Guide for making 2510-02 connector

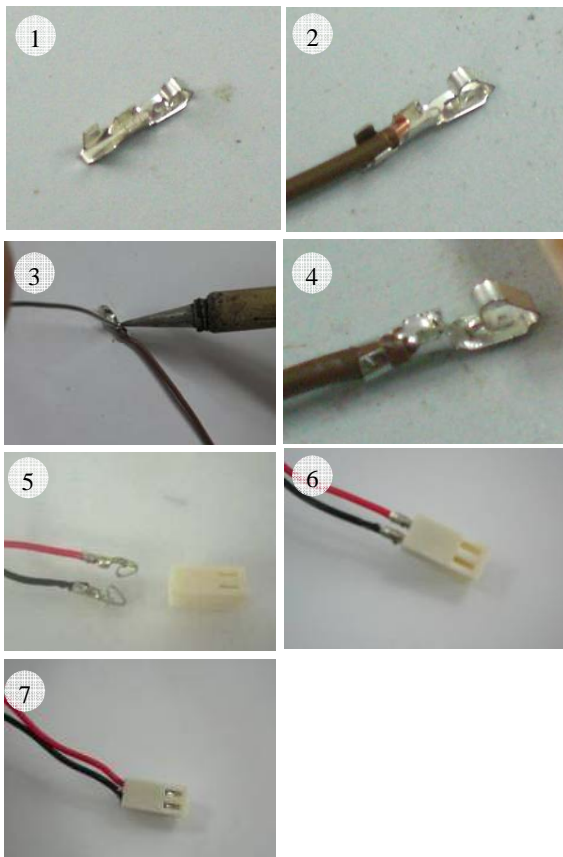


Figure 18

### Guide for making 2510-02 for 9V battery connector:

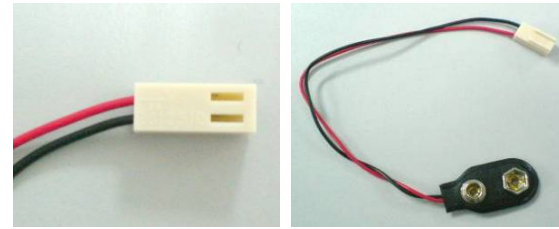


Figure 19 (not included in DIY project set)

### Door lock connector:

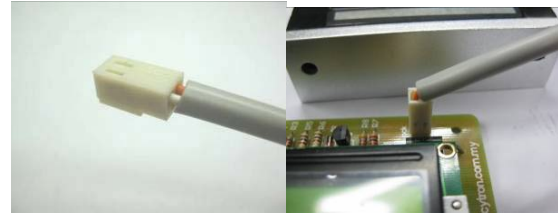


Figure 20

### Guide for making 2510-04 for RFID tag reader connector:

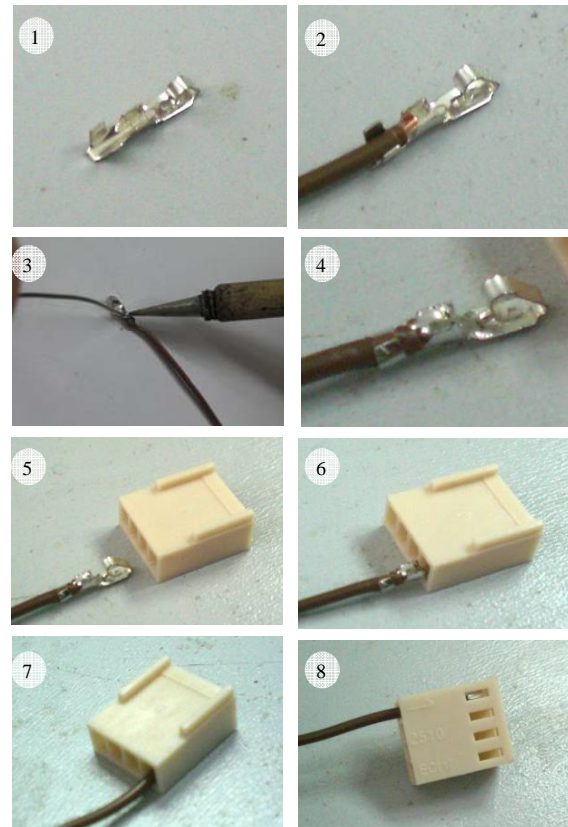


Figure 21

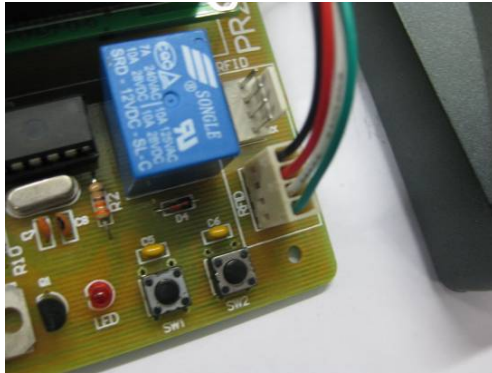
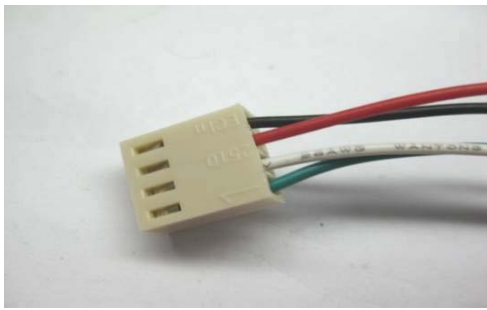


Figure 22

4. Please download the necessary files and document from Cytron Technologies website. These included documentation, sample source code, schematic, component list and software.
5. The next step is to install MPLAB IDE and HI-TECC C PRO into a computer. The MPLAB IDE and HI-TECC C PRO can be downloaded from [www.cytron.com.my](http://www.cytron.com.my). Please refer MPLAB IDE installation step document to install the software. The documents can be used to any version of MPLAB IDE software.
6. After the installation complete, open the project file provided using MPLAB IDE. Please refer MPLAB Open Project document to open the sample program.
7. **Please modify the tag ID in the program same as the tag ID that you want to read.** This step is to allow the system to remember the ID.

Note that the first member of id array initialized in the program is id1=127 and id2=22449, to change the initialized id1 and id2, you can reset the value at the beginning of the program.

```
void main(void)
{
    unsigned char i, repeat, b, database;
    unsigned char data[26];
    unsigned char convert1=0; //cle
    unsigned int convert2=0; //cle
    unsigned char mode=0; //cle
    unsigned char id1[2]={127}; //ini
    unsigned int id2[2]={22449}; //ini
}
```

Figure 25

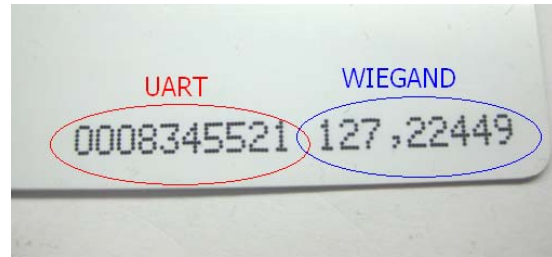


Figure 26

The ID number circled in red is the number that will be displayed if UART RFID reader is used (not used in this case), while ID number circled in blue is the number that will be displayed if Wiegand RFID reader is used. RFID tag is not included in this DIY project set.

In sample source code provided, we **did not use** UART for RFID Read and Display. For more information on UART setting, please refer to previous PR or refer to the PIC16F876A datasheet in microchip website at: <http://www.microchip.com>

8. Plug in power supply for the circuit. User can choose to use battery or AD to DC adaptor.

### AC to DC adaptor:



Figure 23 (Not included in DIY project set)

### Connection to the PCB board:

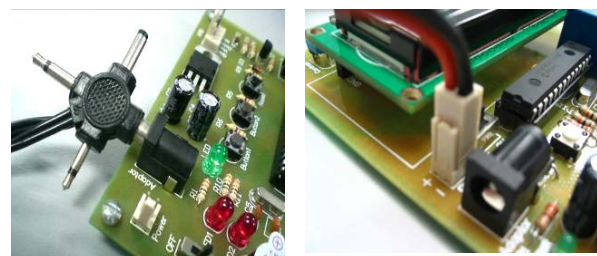


Figure 24

9. Build the project and load the hex file into the PIC microcontroller using the USB In Circuit Programmer (UIC00A). When user build the project, MPLAB IDE will generate hex file. The hex file generated from MPLAB IDE will be named according to project name, not C file name. Cytron Technologies also provide hex file for user. Do not forget to switch ON the power. The programmer is not included in the hardware set but it can be found at Cytron website. (User manual is provided at website).
10. Test the functionality of the PCB board.
11. Have fun!

## TEST METHOD

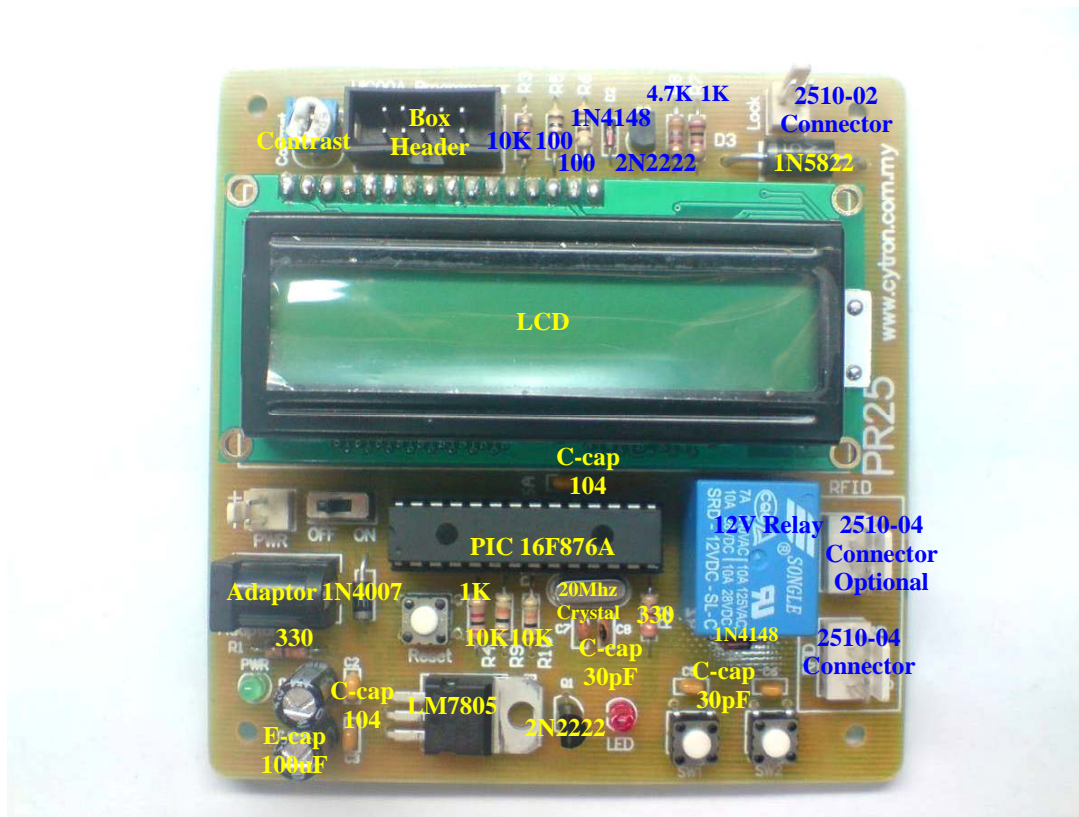
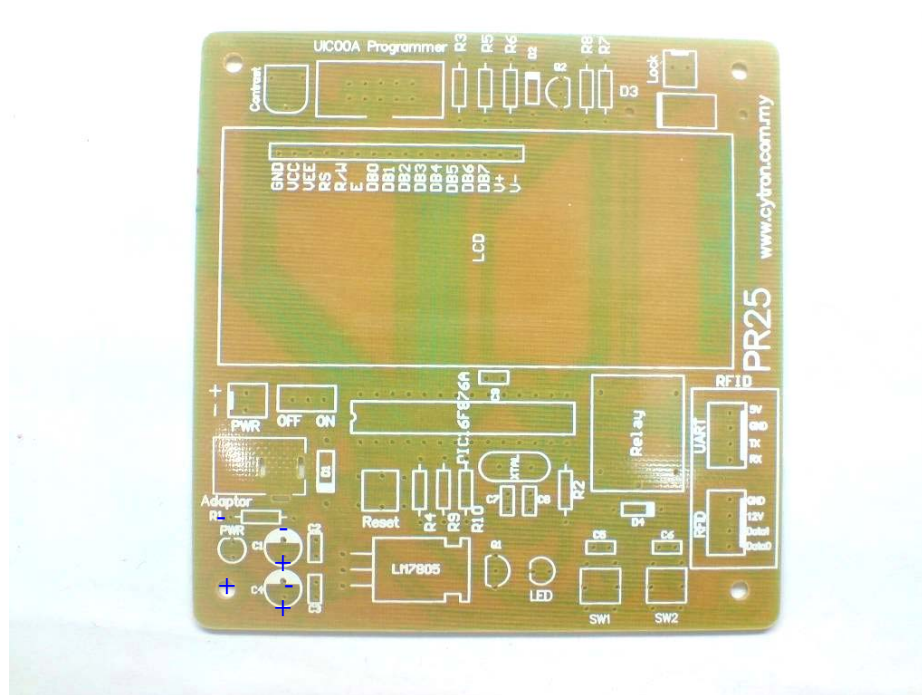
1. Switch ON the power
  - Power LED (Green) will turn ON.
  - Power LED (Red) is OFF.
  - Door lock is locked
  - LCD will show “RFID door security”
  - After a few second it will change to “Place your ID on the reader”
2. Place ID tag on RFID tag reader
  - LCD will display “ID no: xxxxxxxx user not found”. RFID card given is different.
  - Power LED (Red) will turn ON.
  - After a few second it will change again to “Place your tag on the reader”
3. Press SW1
  - Enter to save mode
  - LCD will display “User ID saved” when you place your ID tag on the reader
4. Place the ID tag saved on the reader again
  - LCD will display “ID identified”
  - Power LED (Red) is OFF.
  - Door lock on
  - After a few second it will change again to “Place your tag on the reader”
5. If all steps mention above can be executed, your project is done successfully. Congratulations!!

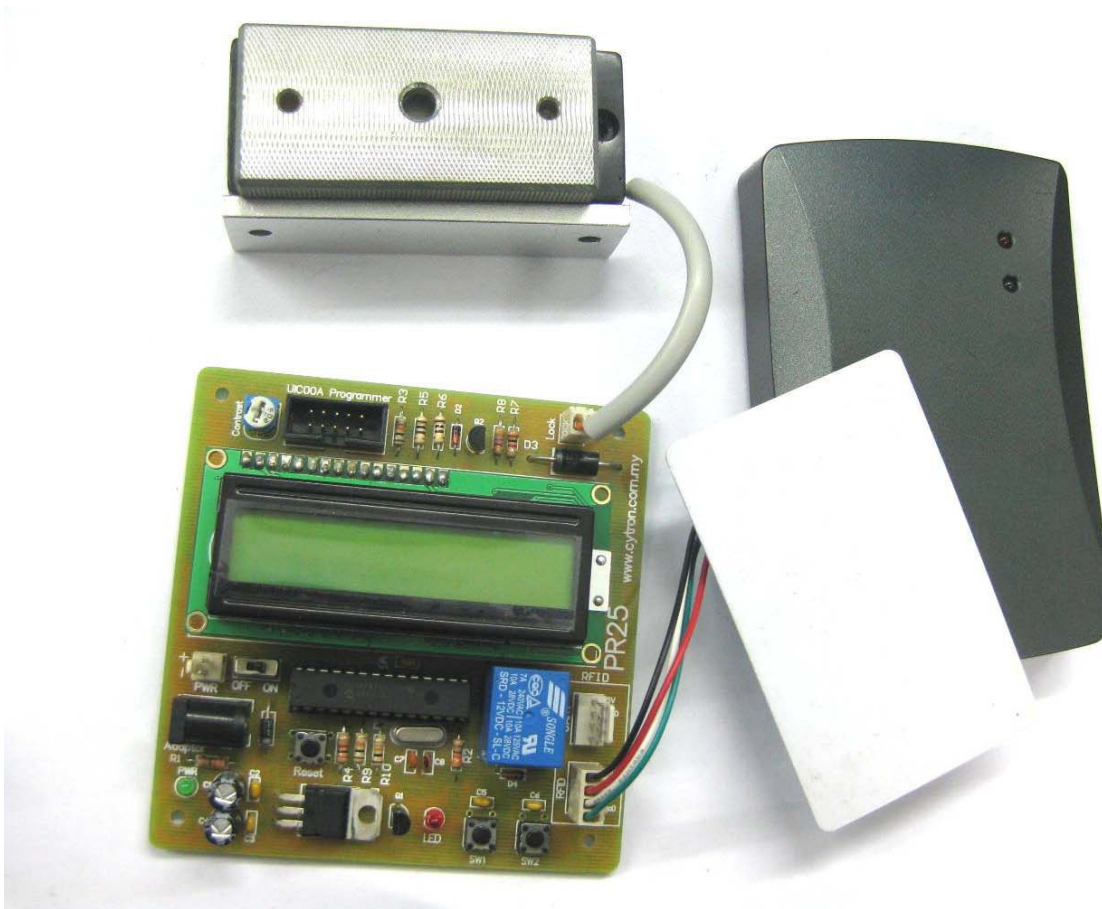
## WARRANTY

No warranty will be provided as this is DIY project. Thus, user is advice to check the polarity of each electronic component before soldering it to board.

## Appendix A

### PCB Layout:





\* Cytron Technologies reserved the right to replace the component in the list with component of the same functionality without prior notice.

*Prepared by*  
**Cytron Technologies Sdn. Bhd.**  
19, Jalan Kebudayaan 1A,  
Taman Universiti,  
81300 Skudai,  
Johor, Malaysia.

*Tel:* +607-521 3178  
*Fax:* +607-521 1861

*URL:* [www.cytron.com.my](http://www.cytron.com.my)  
*Email:* [support@cytron.com.my](mailto:support@cytron.com.my)  
[sales@cytron.com.my](mailto:sales@cytron.com.my)

