**Version n° 1.1**

# LaurTec

# Freedom
# PIC Embedded System

**Author :** *Mauro Laurenti*

**email :** info.laurtec@gmail.com

**ID:** PJ10001-EN

## License

The usage of any project, tutorial, software given by Mauro Laurenti (the Author) imply the acceptance of the following license.

The projects, tutorials, software (material) supplied herewith by Mauro Laurenti is intended for use solely and exclusively for personal use.

All the material is owned by the Author, and is protected under applicable copyright laws.

All rights are reserved.

Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws (Italian or International ones), as well as to civil liability for the breach of the terms and conditions of this license.

Commercial use is forbidden without a written acknowledgment with the Author.

Personal or educational use is allowed if the application containing any part of the given material doesn't aim to commercial use or monetary gain of any kind.

ALL THE MATERIAL IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS MATERIAL. THE AUTHOR SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

# Table of Contents

## Abstract

Freedom is an embedded system mounting 40-pins PIC microcontrollers, developed to suit a wide range of applications, without additional external hardware. Nevertheless, Freedom is a very small system. In 8.3cm x 10.7cm host a 40-pins PIC microcontroller with the possibility to use the following hardware:

- I2C EEPROM memory.
- Real Time Clock/Calendar.
- RS232 Interface.
- RS485 Interface.
- Alphanumeric LCD with contrast trimming.
- Voltage regulator for servo motors.
- Photocell to detect the light intensity.
- Temperature sensor.
- Ultrasonic sensors
- Reset switch.
- On board 5V power supply with LED indicator.
- On board programming connector.
- Each Port is connected to an ICD socket.


## Project Description

In Figure 18 is shown the entire schematic of Freedom embedded system. Even if the schematic seems to be quite big, is not complex at all.

As any complex system is possible to divide it in subsystems which can be easily understood. In this case many subsystems are identified by a single IC. This make the system understanding much simpler but it actually hide the complexity behind each IC.

Freedom embedded system represents a good examples of how is possible to get complex systems just assembling different IC with specif functionalities. The major designer role is to make sure that the electrical specification of each IC is compatible with each other.

Within Freedom is possible to identify the following subsystems, that will be analyzed one at a time:

1. 5V Power supply.
2. Servo motors voltage regulator
3. Transmission Protocols
4. I2C peripherals
5. Ultrasonic sensor
6. Temperature sensor
7. Light sensor
8. Other Hardware
9. Extender ports

## 5V Power supply

Freedom gets its power from an external power supply which could be AC or DC power supply, since the AC-DC converter is assembled on the board. As shown in Figure 1 the power supply is made of four 1N4004 1A diodes, one 7805 linear voltage regulator, filter capacitors, a diode connected to the BATTERY connector plus a LED.

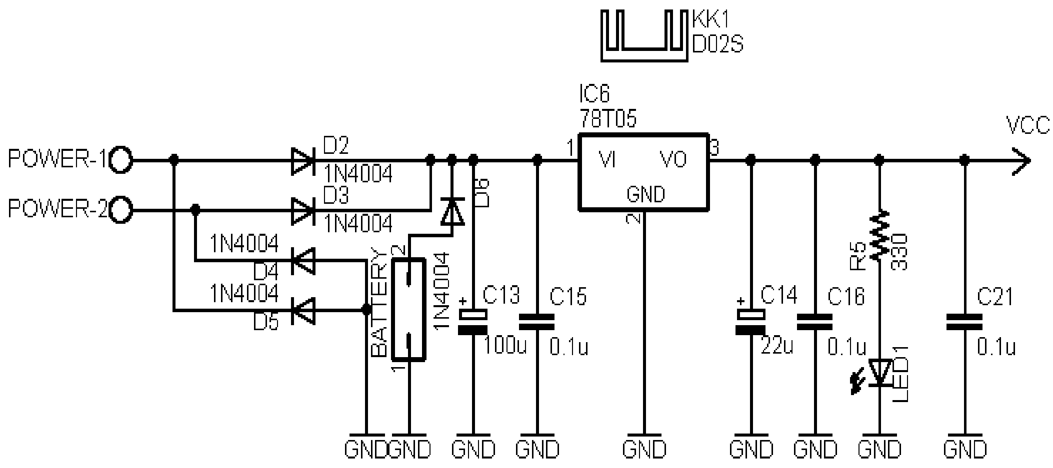



**Figure 1:** *Power supply schematic*

As mentioned, Freedom requires an external power supply so no transformer is included. The reason behind this choice is related to the fact that the board can be used within a more complex system from which is possible to get the power. Moreover the transformer size would also depend on the hardware that is connected to Freedom, which could require more current than Freedom itself.

The four diodes allows Freedom to be powered using AC and DC power supply. If the power is AC the voltage must be within this range 6Vac-9Vac. If the power is DC, the voltage must be in the range of 8V-12V. Standard DC power supplies, stabilized or not, with a voltage range of 9V-12V can be used without any problem. For the current that the power supply must be able to withstand, at least 300mA is recommended (if no external hardware is connected to the Extender connectors). By the way, in all cases, the current must be not lower than the current sink by Freedom plus the connected hardware.

In case a DC power supply is used to power Freedom, you do not have to worry about the wiring polarity from the power connector. In fact the rectifier diodes will properly correct the polarity letting the 7805 voltage regulator working without problems, since it will always get the positive voltage as input[1].

After the diodes the linear voltage regulator 7805 will stabilize the voltage to 5V. For properly working the 7805 requires a minimum input voltage of 7.5V DC. The connector where you must connect the external power supply is labeled POWER as shown in Figure 4; this connector is just next to the connector labeled SERVO[2]. To get low ripple output voltage, there are the input and output capacitors. These allow a reduced ripple and a better transient response. On the output side there is also a red LED for showing that the 5V power supply is on[3].

The heat sink connected to the voltage regulator could or not be required, depending on the required

1 For more information I would recommend reading the project "Stabilized power supply 5V-12V, 1A". You can download it from www.LaurTec.com

2 The main and the servo power have a common ground. Moreover the ground layer is also connected to the screw holes located on the board corners.

3 It is important to point the fact that the LED could be on even if the 7805 is not properly working.

current and the input voltage, refer to the 7805 datasheet for more information about the heat sink requirements. As already mentioned, next to the voltage regulator input, there is a connector labeled BATTERY, its location is shown in Figure 2. This connector is used to connect an external battery that can be used as energy buffer.
This feature is particularly handy for the Real Time Clock/Calendar installed on board, avoiding to loose the clock time information in case of main power problems.
In fact, till there is the main power the diode D6 is interdicted (the cathode is more positive than the anode[4]) and the battery is not used.

Due to the rectifier diodes, the battery will power just Freedom and not the boards connected before it. All the board connected directly to Freedom using the PORT connectors, will be powered as well.

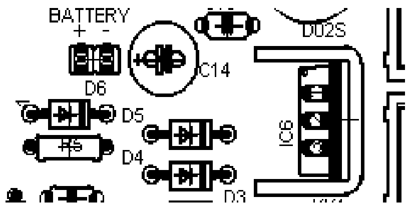



**Figure 2:** *BATTERY connector location*

[4] For properly using this feature the minimum input voltage is 12Vdc or 9Vac, otherwise the battery will be discharged.

## Servo motors voltage regulator

Freedom is an embedded system which could be used for automation systems where Servo motors are required. Servo motors are small motors with internal gears and controlling electronics for designing simple and powerful systems. Servo motors are frequently used in robotics applications since they have very high torque of severals Kg/cm. In Figure 3 is shown the schematic related to the Servo motor section.

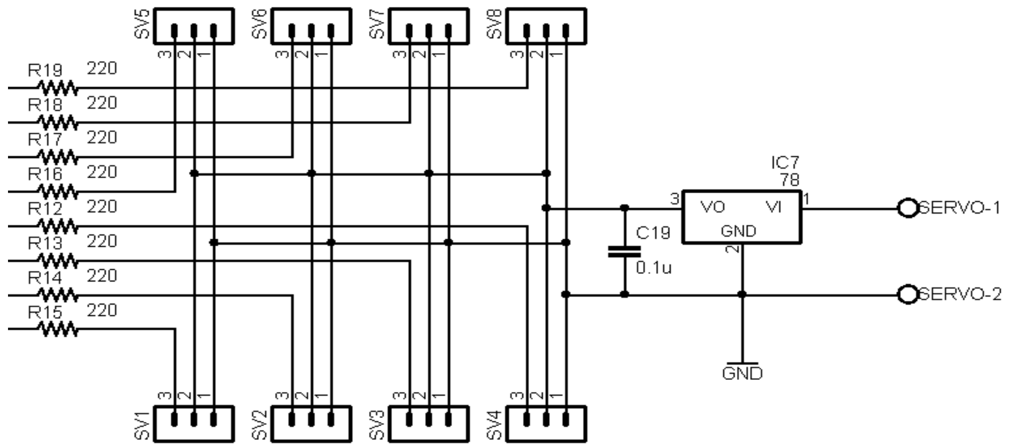



**Figure 3:** *Servo Motors schematic*

It's possible to understand that the motors are connected using 3 pins connectors, the same one that you can find as jumpers on the PC main board. The Servo power supply is generally between 5V and 6V. The linear regulator labeled 78xx could be the LM7805 or the LM7806 depending on the requirements[5]. The Servo motor get powered using the connector labeled SERVO shown in Figure 4. From that Figure is also possible to read the wiring polarity which is required (this polarity must be respected).

Since different linear regulator could be used, the minimum voltage that can be used will depend on the voltage regulator itself. For instance if you use the 7805 the minimum input voltage will be 7.5V[6].

The presence of IC7 is due to the fact that it keeps the main 5V power supply for the PIC decoupled from the Servo power supply. The Servo motor can sink 100mA-150mA but it can easily require current peak 5 times bigger, this transient could create voltage droop that could reset the PIC if the power is not properly decoupled.

Since the voltage regulator is a 78xx which can withstand 1A (Some version can withstand 1.5A-3A) all the current sinked by the active Servos must not be greater than 1A.

Freedom can support up to 8 Servo Motors as shown in Figure 4, since the current limit is about 1A, in case more Servo motors must be used, you must remove the voltage regulator (creating a jumper between the pad 1 and 3) and use an external power supply suitable for powering the Servo motors.

[5] If you are using a different Servo you can change the voltage regulator with another ones pin to pin compatible.

[6] For more information about the min and max voltage that can be dissipated by the IC, I would recommend reading the datasheet. Depending on the voltage regulators these parameters could be different.

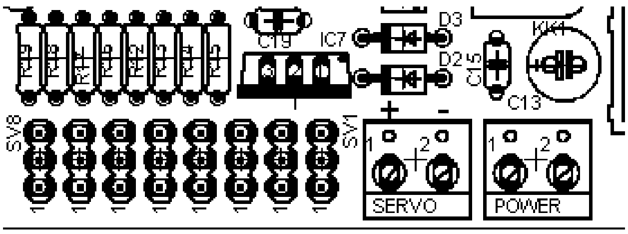



**Figure 4:** *Servo Motors connectors view*

The control pin of each servo is connected to the PORTD of the PIC through a protection resistor (the protection is against inverting the pin 1 and 3).

Generally the color coding used for the servo wires is Black for the negative power supply, Red for the positive power supply and white or yellow for the controlling wire.

From Figure 3 and Figure 4 is possible to see that the ground corresponds to pin 1 (black wire), the central pin is the positive one (red wire) and the middle one is the controlling wire (yellow wire)[7].

[7] The pin location is not standard. Make sure that the Servos you are using are compatible with the pin out just described.

## Transmission Protocols

Any kind of embedded systems require a bus for communicating with other systems. Freedom as other commercial embedded systems, supports the RS232 and the RS485 serial buses. These two buses can not be used simultaneously. Beside these two protocols the I2C and the SPI are directly supported[8].

The RS232 protocol allows ease connection with any PC that supports that protocol. The RS485 has been thought to allow the connection between multiple systems which are distant from each other. The RS485 uses a differential bus, with only two lines is possible to connect multiple systems[9]. In Figure 5 is shown the schematic related to the RS232 and RS485 connections.

The voltage level shift from the TTL level (0V-5V) to RS232 is made by the MAX232. This device is using a pump charge, made with the 4 external capacitors, this allow to get the ±15V required by the RS232 protocol[10].

For the bus RS485 the required level shift is made using the MAX481. The termination resistors R6 and R7 they generally have a value 10KΩ or so, but they can be changed depending on the application requirements.

The RS232 output is made available using the standard DB9 female connector using the same standard pin out used by the PC[11].
As alternative to the DB9 connector, it is possible to use the 10 pins ICD connector (in Figure 5 labeled RS232). The TX, RX, GND pins are connected to the same pins used for the DB9 connector.

The RS485 output can be taken directly from a simple 2 pins connector. Form the schematic is possible to see that the RS485 requires an additional control pin (the PIC pin RB1 has been used) which allows to select the communication direction RX or TX[12].

As mentioned before, the two protocols can not be used at the same time, since they share the PIC hardware[13]. The choice between one or the other is made using the jumpers labeled TX and RX, that allow to connect the TX pin (RC6) and the RX pin (RC7) to the MAX232 or the MAX481. In Figure 6 are shown the RX and TX jumper settings for an RS232 transmission. In Figure 7 are shown the RX and TX jumper settings for a RS485 communication.

---

8 Which protocols are actually available depends on the PIC which is used. Normally the I2C and the SPI can not be used at the same time. Furthermore additional serial bus can be supported if external transceiver are used. As example refer to the Project PJ30001-EN "RS232 CAN protocol converter".

9 The RS485 is a quite robust protocol thanks to the differential data transmission. It is normally used within environment where data integrity is particularly important, such as industrial environment.

10 For additional information about the RS232 protocol, reading the Tutorial "RS232 protocol explained" is recommended (refer to the Bibliography).

11 As shown by the schematic, just the GND, TX, RX pins are available. The other handshaking signals are not used.

12 A transmission protocol that support just TX or RX one at a time, is called half duplex. A transmission where TX and RX can be done at the same time, is called full duplex. The RS232 is an example of full duplex protocol, the RS485 is an example of half duplex protocol. It is worth to mention that there are variation to the RS485 protocol that support full duplex transmission. The variation requires, by the way, doubling the buffers and the wiring.

13 There are PICs that have multiple USARTs, but Freedom does support just one.

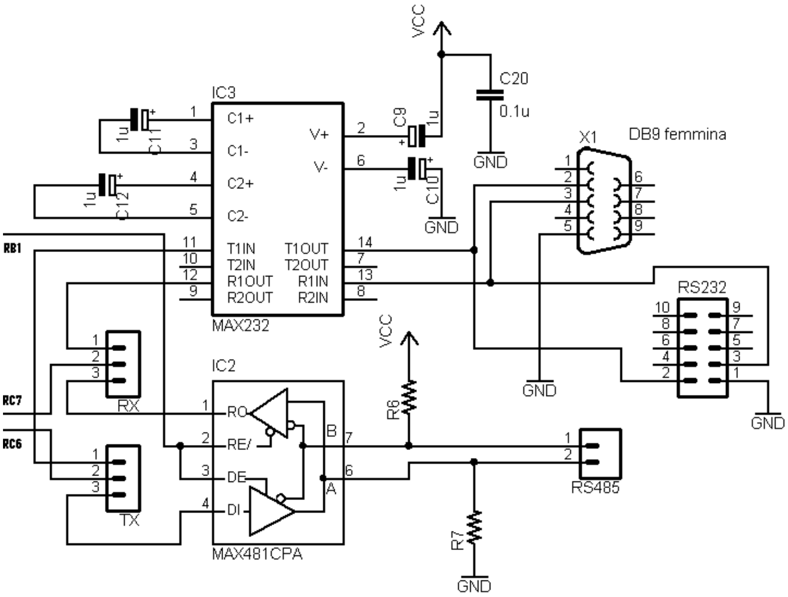



**Figure 5:** *RS232 e RS485 schematic*

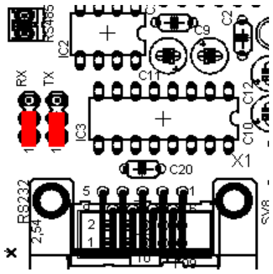


**Figure 6:** *RX-TX jumpers settings for a RS232 transmission*

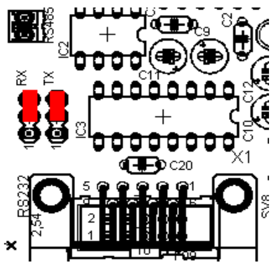


**Figure 7:** *RX-TX jumpers settings for a RS485 transmission*

## I2C Peripherals

On the Freedom board there are two peripherals already connected o the I2C bus[14]. Those are the EEPROM 24xxx and the Real Time Clock/Calendar PCF8563P as shown in Figure 8.
Thanks to the EEPROM is possible to store additional data beside the PIC internal EEPROM or flash memory. The Real Time Clock/Calendar is used as system clock with alarm capability.

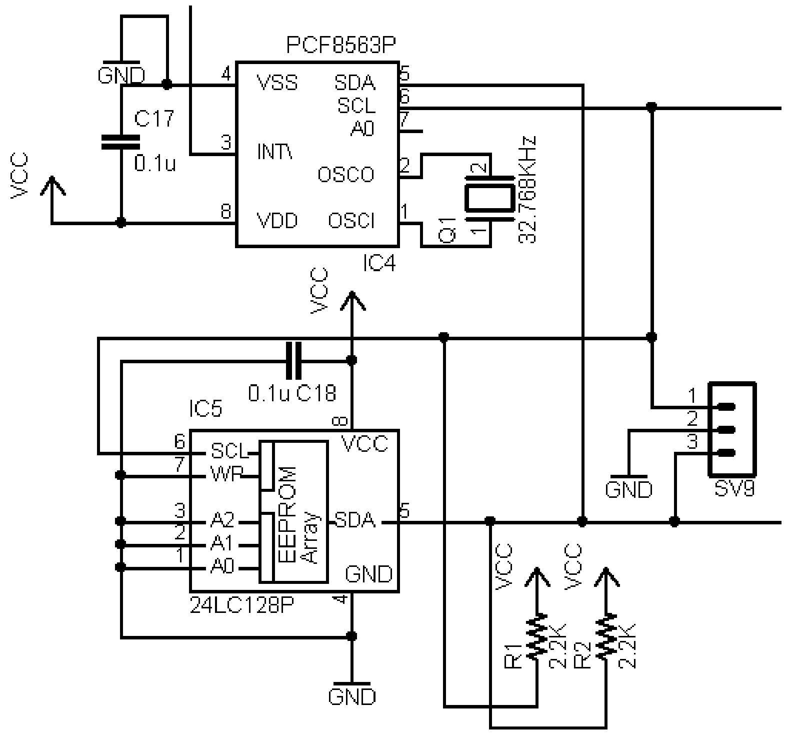



**Figure 8:** *EEPROM and Real Time Clock/Calendar schematic*

The I2C bus is represented by the RC3 and RC4 of the PIC (different pins are used for the PIC which have USB controller), the bus lines are connected to Vcc through the pull-up resistors R1 and R2. The pull-up resistors are required since the line SDA (Serial Data) and SCL (Serial Clock) are open collector signals.

Using the SDA and SCL bus from the PORTC connector, allow to connect further devices on the bus, extending Freedom capabilities. Beside the PORTC connector, Freedom is designed to support ultrasonic sensors directly connected to the I2C bus.

It is worth to point, that the external I2C peripherals must not create address conflict with the IC already mounted on board. Moreover all the devices must be capable of working with 5V power supply. A summary of the addresses already used on board are shown in the Table 1.

[14] For more information about the I2C protocol, reading the Tutorial "Bus I2C explained" is recommended (refer to the Bibliography).

| IC | Read address | Write address |
| --- | --- | --- |
| **24LC512** | A1 H | A0 H |
| **24LC256** | A1 H | A0 H |
| **24LC128** | A1 H | A0 H |
| **24LC64** | A1 H | A0 H |
| | | |
| **PCF8563P** | A3 H | A2 H |

**Table 1:** *Used Read/Write addresses on the I2C Bus*

Since many EEPROMs are pin to pin compatible Table 1 shows more EEPROMs. Some EEPROMs have external pins labeled A0, A1, A2; these pins are used to set an external address that allows connecting up to 8 similar devices, setting different values for A0,A1,A2. Freedom has this pins connected to ground, so A0=0, A1=0, A2=0. If other IC memory are required, different address must be used. The Real Time Clock/Calendar has a wired address with no external address pins, this means that just one Real Time Clock/Calendar can be used within the same I2C bus section[15].

The Real Time Clock/Calendar, as many quartz clock, uses an external 32768Hz quartz as time reference for generating one second ( 1Hz ) beat.

The PCF8563 can be set with an alarm time, that generates an interrupt on the event. The PCF8563's interrupt output pin is connected to the PIC INT (RB0) pin with a pull-up resistor, creating a wired-or connection. This connection topology allows multiple interrupt pins to be connected to RB0. The device interrupt pins must be open collector or open drain (push pull topology can not be used).

As brief summary is worth to point that, in case you want to use the PCF8563 you must keep the I2C bus (RC3 e RC4) and the pin INT (RB0) without further incompatible features. If you want to use just the EEPROM you can use the pin INT (RB0) for other functions, but you must keep in mind that the pin has a pull-up resistor. For more information about the PCF8563 and the EEPROM just mentioned, is worth to read the datasheet.

As shown from the schematic in Figure 8 the I2C bus is connected to SV9 connector, for ease connection with other boards (master and slave connections) or peripherals. The SV9 pin out is shown in Table 2.

| SV9 Pin | I2C functionality |
| --- | --- |
| 1 | SCL |
| 2 | GND |
| 3 | SDA |

**Table 2:** *SV9 pin out*

15 The I2C could be divided in sections. Within different section you could have same addresses.

## Ultrasonic sensor

Ultrasonic sensors can be very handy in automatic systems such as robotic applications, for measuring distances. The way it works, is the same one used by bats in nature, an ultrasound pulse is transmitted and the time for receiving the echo is proportional to the distance from the pulse source and the object that has generated the echo. Depending on the resolution and the size of object that you want to detect, a different ultrasonic type sensor must be chosen.

In some applications it could be handy having more sensors watching different areas. In these kind of applications, to save microcontroller I/O pins, it could be useful a network of sensors connected on the I2C bus, instead of using trig/echo sensor type, which would requires two pins per sensor (trig and echo pin[16]). Freedom allows with easy setting steps, to use I2C and Trig/Echo sensor types.

In Figure 9 is shown the SRF05 ultrasonic sensor with Trig/Echo interface.

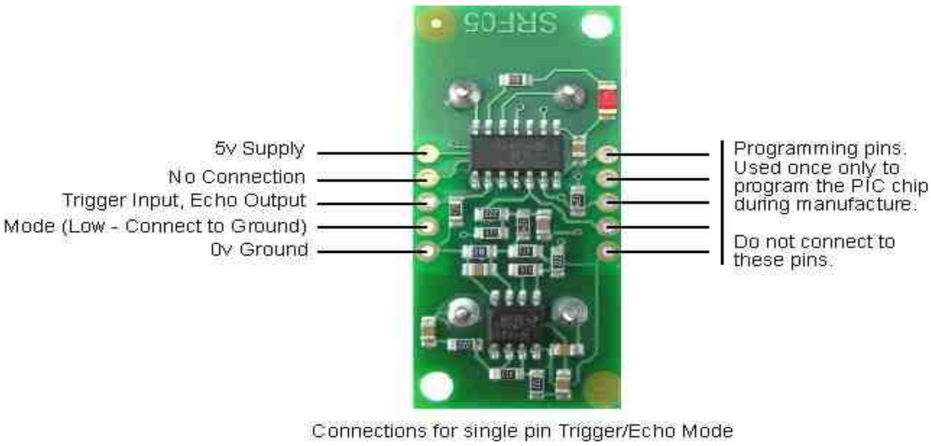



**Figure 9:** *Ultrasonic sensor type SRF05 with Trig/Echo interface*

In Figure 10 is shown the SONAR connector where it is possible to connect the sensor/s. As shown the connector is compatible with the SRF05 sensor. Pin 1 is connected to +5V and can be used as reference to check the connection equivalence.

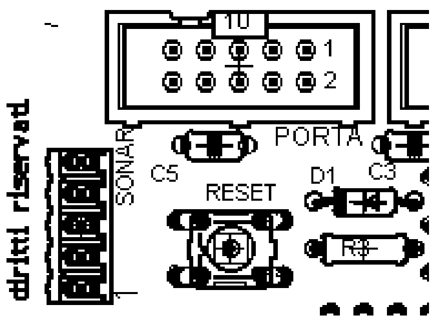



**Figure 10:** *Ultrasonic sensor connector, +5V is pin 1*

For setting Freedom on Trig/Echo modality, the TRIG, ECHO jumpers must be set as shown in Figure 11.

16 Trig and Echo pins are used to Trig the ultrasonic pulse and to measure the Echo width. The Trig pin could be in common, but sensor interference could be created due to multi-path reflections.

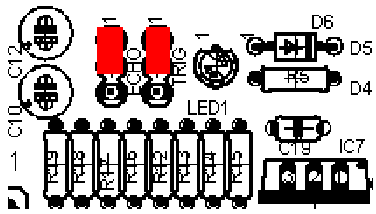



**Figure 11:** *Jumper setting for Trig/Echo modality*

The pin Trig, from the sensor, is connected to the RC0 pin of the PIC while the Echo pin is connected to RC5. For setting Freedom on I2C modality, the TRIG, ECHO jumpers must be set as shown in Figure 12. In this modality the sensor pin SDA is connected to RC4 while SCL is connected to RC3.

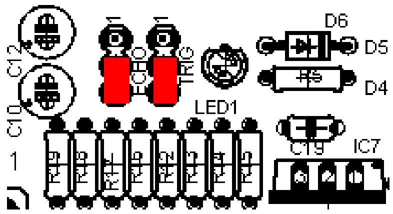



**Figure 12:** *Jumper setting for I2C modality*

## Temperature sensor

The temperature sensor mounted on board allows Freedom to read one of the main environmental characteristic. In Figure 13 is shown the schematic sketch related to the temperature sensor DS1820. The sensor DS1820[17], with 1-wire output, has been chosen against sensors with analog output, since it allows multiple sensors on the same bus, connecting them in parallel.

Since the sensor output is open drain[18], a pull up resistor is required (R10). The sensor output is connected to RA5[19] which must be set as I/O for making the communication. Each digital sensor has a unique address[20], so, as already mentioned multiple sensors can be connected in parallel. It will be up to the firmware to select the sensor of interest. If multiple sensors will be connected, just one pull-up resistor is required any way, since R10 will connect to Vcc all the outputs

Many programming languages has already libraries to properly read such kind of sensors without headache. Nevertheless is important to read the DS1820 datasheet to get more information about the commands and addressing modalities.

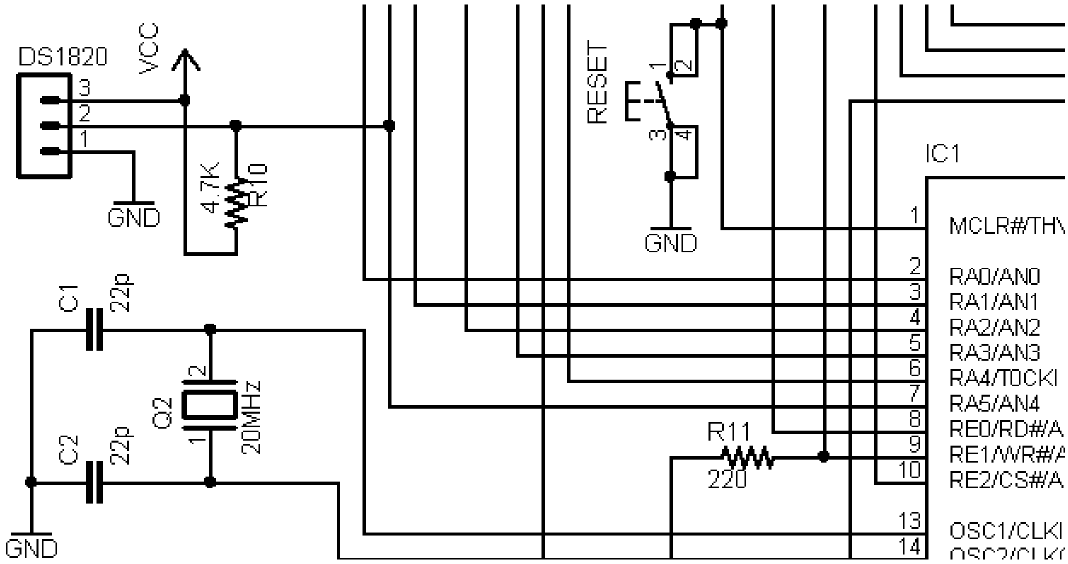



**Figure 13:** *DS1820 schematic sketch*

[17] This sensor is manufactured by Maxim/Dallas. The DS18S20 has been recently released as replacement for the DS1820.

[18] For more information, reading the article "1000 questions 1000 answers" is recommended (refer to the Bibliography).

[19] Since the output sensor is connected to a pin which can be configured as analog input (beside digital input), the temperature sensor LM35 can be also used. In this case the pull-up resistor must be removed for proper output reading. It is worth to point the fact that if the LM35 is used, just one sensor can be connected.

[20] The concept of 1-wire addressing is analogue to the I2C addressing.

## Light sensor

The light sensor is a photo resistor labeled FR, it is connected as voltage divider[21] with a 100KΩ resistor, as shown in Figure 14.

The chosen photo resistor has a resistance in the darkness of 100KΩ, this means that in this condition the output voltage is 2.5V[22]. If the light increases the photo resistor decreases the resistance, causing an increase of the output voltage (since the voltage is measured on R8). So increasing the light the voltage will increase, but it will be always between 2.5V and 5V. Mathematically the equation that explain what we have described is the following:

$$V_{OUT} = \frac{R_8}{R_8 + R_{FR}(i)} \cdot V_{CC}$$

where:

$R_8$ = 100KΩ
$R_{FR}$ *(i)* = Photo resistor impedance as function of the incident light (min. 10KΩ max. 100KΩ)
$V_{CC}$ = 5V

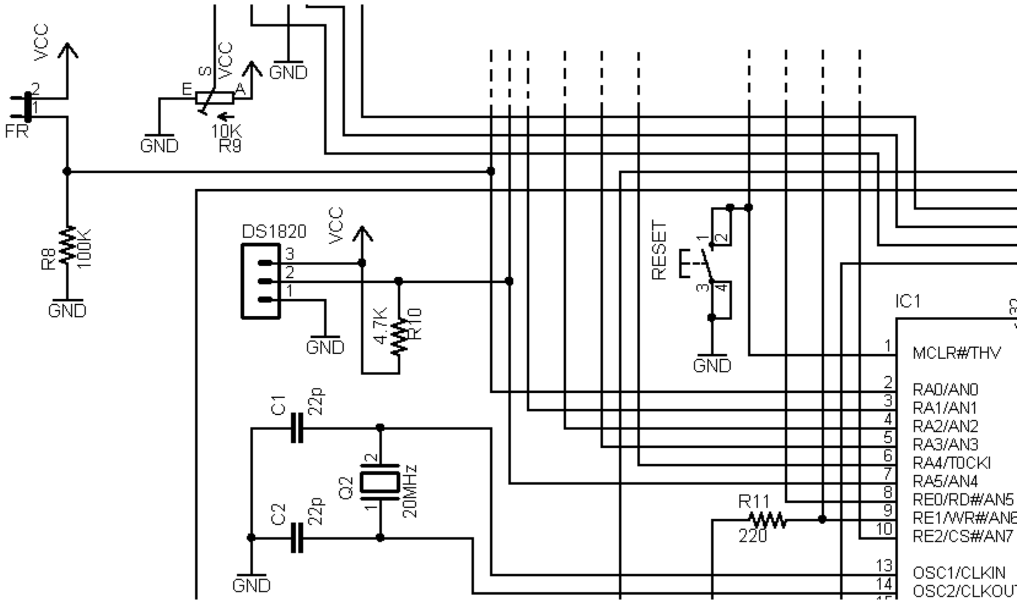



**Figure 14:** *Light sensor schematic sketch*

By the way this equation can not be actually used, since the function of RFR(i) is not known, unless you use a professional and expensive sensor. A workaround would be experimental measures using known light intensity. The voltage divider output is monitored through the RA0 PIC analog input. No buffer has been used to keep the over all hardware simpler. The hardware just described could be used within a robot or domotic applications, to discriminate the day and the night. A wire could be required to locate the photo resistor in a better place far from the board itself. By the way is better to avoid length longer than 2-3m, since the analog input is not protected against ESD and electrical shock in general.

[21] Per ulteriori informazioni in materia si rimanda al Tutorial "Teoremi ed applicazioni fondamentali dell'elettrotecnica" (refer to the Bibliography).

[22] Questi valori sono validi per il fotoresistore utilizzato, il quale possiede una resistenza al buio di circa 100KΩ, mentre una resistenza minima sotto illuminazione di 10KΩ.

## Other peripherals

In addition to the mentioned hardware, there are further hardware and utilities that make Freedom quite versatile.

- RE1 pin is connected through a 220Ω resistor to a buzzer. The buzzer could be replaced with a LED as well, keeping the same buzzer polarity constraints. This feature is quite useful for any acoustic or visual signal. For some buzzer the resistor must be replaced with a jumper to get enough power. This is not true for the LED, which must have always the resistor, limiting the current out of the microcontroller.
- The RESET button, allows the system reset at any time there is a software problem or you want just reboot the system for any other reason, this button is labeled RESET. If the microcontroller programmer that is used does not have any current limiting protection, this can be damaged if you press the RESET button during programming. This happen due to the fact that the pin MCLR get shorted to ground.
- LED1 is used for showing that the power supply is connected[23].
- The pin INT/RB0 has a pull-up resistor for a wired-or connection. This pin is used for the Real Time Clock/Calendar but further peripherals can be connected.
- The LCD connector allows a direct connection of a standard LCD alphanumeric display, using 4 bits interface modality. Since the LCDs generally have 16 pins, it means that some of the pins will not be connected. Freedom has the trimmer which is used to regulate the LCD contrast. The LCD connector pin out is written in Tabel 3. If you are programming the PIC directly on board, it is recommended to disconnect the LCD because it could create programming failures, due to current limitations of some programmers.

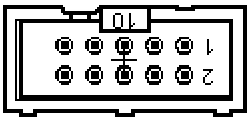
| LCD Connector pin | Usage | Standard LCD pin |
|---|---|---|
| **1** | GND | GND |
| **2** | Vcc | Vcc |
| **3** | Contrast regulation | Contrast regulation |
| **4** | RD2 | RS |
| **5** | GND | R/W |
| **6** | RD3 | E |
| **7** | RD4 | DB 4 |
| **8** | RD5 | DB 5 |
| **9** | RD6 | DB 6 |
| **10** | RD7 | DB 7 |

**Table 3:** *LCD Connector pin out and the LCD/microcontroller connections*

[23] It is wort to mention that, if the LED is ON does not mean that the voltage is properly set to 5V but that the voltage and the current provided by the voltage regulator is enough to turn on the LED.

## Extender ports

All the PIC port pins are connected to 10 pins ICD connector, which is providing also the ground and the +5V. This connectors are quite useful for connecting external hardware to Freedom.
As example the PORTC connector pin out is shown in Figure 15; this pin out is used for all the other ports. The only exception is PORTA e PORTE, these ports don't respect the pin out since they do not have all the pins. For instance the PIC16F877A has only RA0-RA5[24], while PORTE has just RE0-RE2, so there are some unused pins.




**Pin out**

1 : **PORTC0**
2 : **PORTC1**
3 : **PORTC2**
4 : **PORTC3**
5 : **PORTC4**
6 : **PORTC5**
7 : **PORTC6**
8 : **PORTC7**
9 : Vcc
10 : **GND**

**Figure 15: PORTC pin out**

As shown in Figure 19 each port connector is labeled with the name of the port itself. Beside the port connectors there are two further connectors. The first one is the LCD connector, previously described, while the second one is the programmer connector. The programmer connector allows on board PIC programming, so is not required removing the microcontroller each time. This let you saving test time and avoids pins damaging.

In Figure 16 is shown that the MCLR pin is connected to Vcc through the R3 resistor, as recommended by the datasheet. The RESET button connects the MCLR pin to ground, once is pressed. D1 isolates the board power from the programmer power supply during the programming step[25].
In fact during the microcontroller programming, MCLR is about 14V, since the cathode is at higher voltage than the anode, the diode is like an open switch.

[24] This I/O pins are not always the same. Some PIC with internal quartz allows using the OSC pins as general I/O pins (OSC1-> RA7 while OSC2-> RA6).
[25] R3 guaranties anyway a good protection even if D1 is not used (replaced by a jumper).

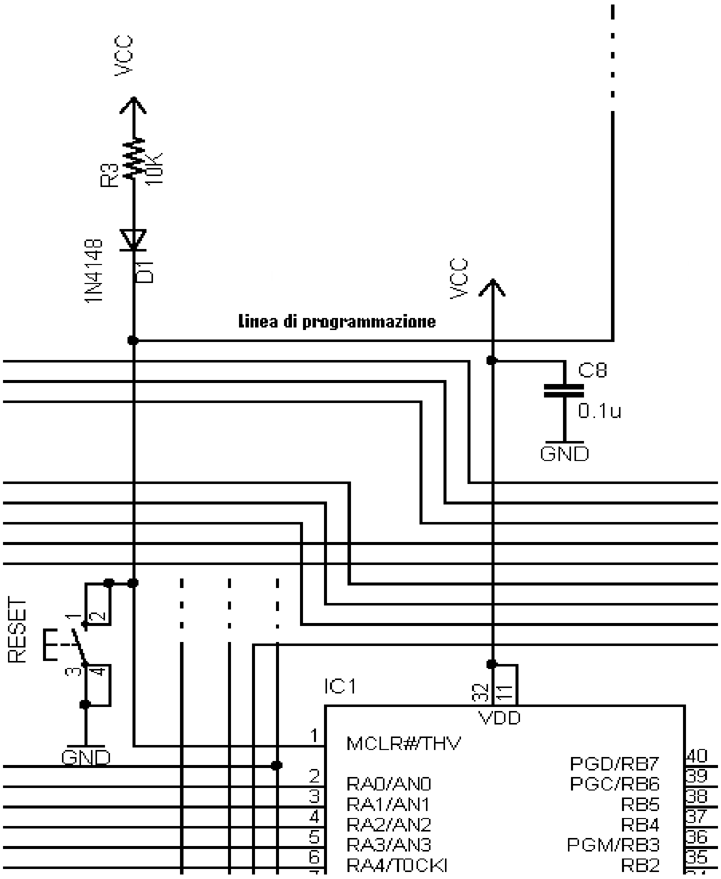
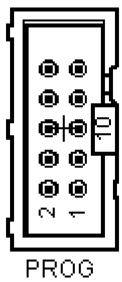



**Figure 16:** *Schematic sketch related to the RESET button.*

The PROG programming connector pin out is shown in Figure 17. During programming is good practice to disconnect any hardware connected to PORTB, in particular to the programming pins RB6 and RB7). External hardware connected to these pins could create programming failures.




**Pin out**

1-2 : **Vcc**
3-4 : **RB6**
5-6 : **RB7**
7-8 : **MCLR**
9-10 : **GND**

**Figure 17:** *PROG connector pin out*

In Table 4 are shown all the PIC connections and the available hardware. It is wort to remind, that depending to the design choice some hardware could be not available.

At the end in Figure 18 is shown the entire Freedom schematic. Now you can understand that the schematic is not hard to understand if analyzed in functional blocks.

| Pin | Hardware | Other Features |
|---|---|---|
| **RA0** | FR, photo resistor input | I/O or analog input if the photo resistor is not used. |
| **RA1** | Available | I/O or analog input. |
| **RA2** | Available | I/O or analog input. |
| **RA3** | Available | I/O or analog input. |
| **RA4** | Available | I/O or TOKI input. |
| **RA5** | Bus 1-wire for DS1820 or LM35. | I/O or analog input if R10 pull-up resistor is removed. |
| **RA6** | Available if internal oscillator is not used (not all the PICs have internal oscillator). | I/O if the internal oscillator is used. |
| **RB0** | Real Time Clock/Calendar interrupt (wired-or) | INT, wired-or. |
| **RB1** | R*/W enable for RS485 | I/O if RS485 is not used, interrupt pin. |
| **RB2** | Available | I/O, interrupt pin. |
| **RB3** | Available | I/O, interrupt pin. |
| **RB4** | Available | I/O, interrupt pin. |
| **RB5** | Available | I/O, interrupt pin. |
| **RB6** | Available | I/O, interrupt pin. |
| **RB7** | Available | I/O, interrupt pin. |
| **RC0** | TRIG ultrasonic sensor | I/O if the TRIG/ECHO modality is not used. |
| **RC1** | Available | I/O (PWM for some PIC) |
| **RC2** | PWM | I/O if PWM module is not used. |
| **RC3** | SCL for I2C bus; used by the EEPROM and the Real Time Clock/Calendar | I/O if the I2C peripherals are not used. |
| **RC4** | SDA for I2C bus; used by the EEPROM and the Real Time Clock/Calendar | I/O if the I2C peripherals are not used. |
| **RC5** | ECHO signal from the ultrasonic sensor. | I/O if the TRIG/ECHO modality is not used. |
| **RC6** | Linea TX per trasmissioni RS232 o RS485 | I/O if neither RS232 nor RS485 hardware is used. |
| **RC7** | Linea RX per trasmissioni RS232 o RS485 | I/O if neither RS232 nor RS485 hardware is used. |
| **RD0** | Servo motor SV1 | I/O in the Servo motor is not used. |
| **RD1** | Servo motor SV2 | I/O in the Servo motor is not used. |
| **RD2** | Servo motor SV3/ RS LCD | I/O in the Servo motor and the LCD are not used. |
| **RD3** | Servo motor SV4/ E LCD | I/O in the Servo motor and the LCD are not used. |
| **RD4** | Servo motor SV5/ DB4 LCD | I/O in the Servo motor and the LCD are not used. |
| **RD5** | Servo motor SV6/ DB5 LCD | I/O in the Servo motor and the LCD are not used. |
| **RD6** | Servo motor SV7/ DB6 LCD | I/O in the Servo motor and the LCD are not used. |
| **RD7** | Servo motor SV8/ DB6 LCD | I/O in the Servo motor and the LCD are not used. |
| **RE0** | Servo motor SV1 | I/O or analog input |
| **RE1** | Buzzer | I/O or analog input if the buzzer is not used. |
| **RE2** | Servo motor SV1 | I/O or analog input |

**Table 4:** Connection Summary about the pin and the available hardware. For more information about the pin functionality, reading the used PIC datasheet is recommended. In the table just the main

functionalities are written.

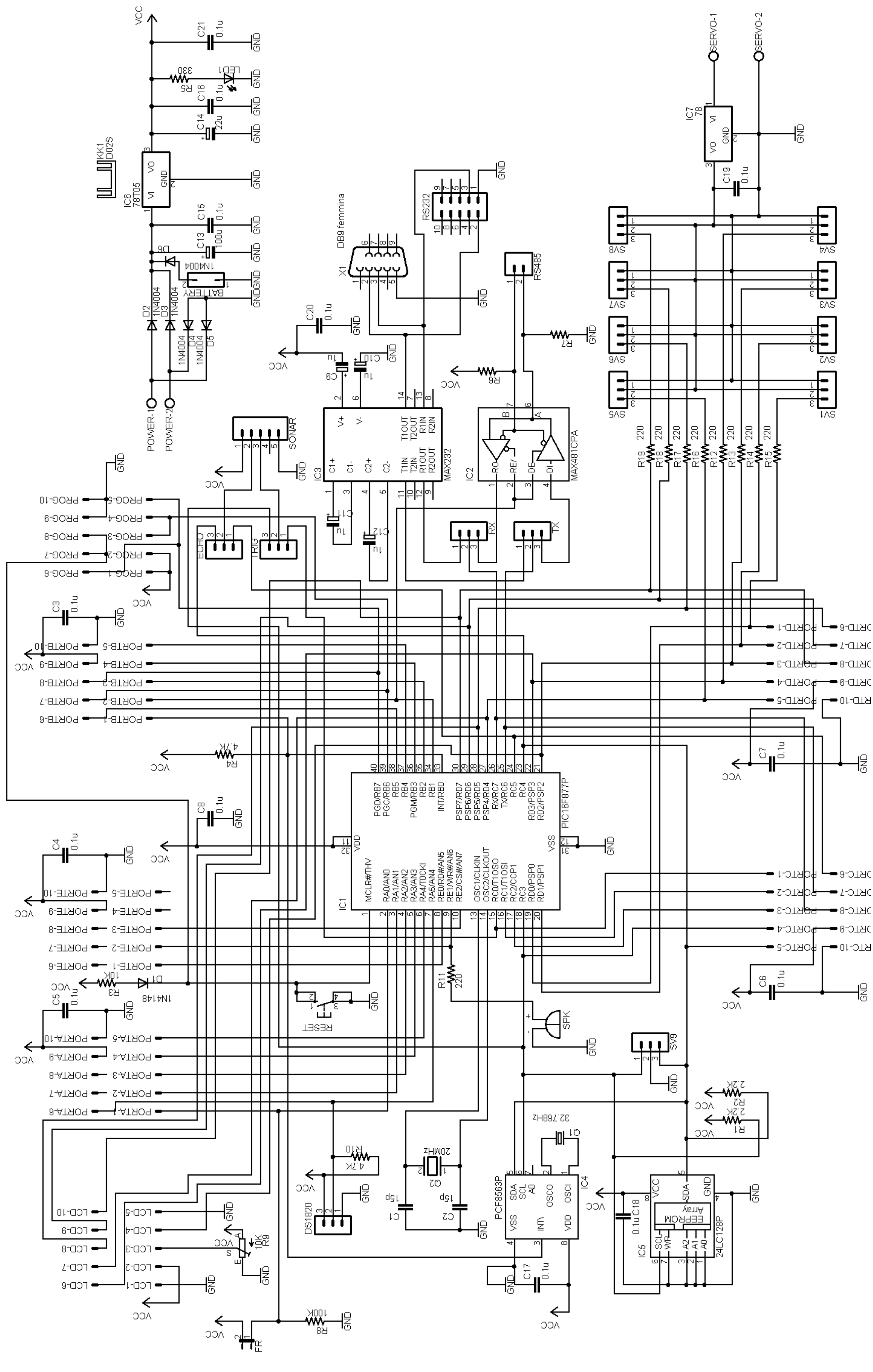



**Figure 18:** *Freedom schematic*

Below there are the components that are used on board. Some notes are important:

- **Used PIC**
  All the PIC with 40 pins could be used but not all the hardware could be available due to different pin out. I have personally used the following PICs : PIC16F877A, PIC18F4580, PIC18F458. PIC with USB port can be used as well but you can not use the I2C bus due to a different pin out (you must do some blue wiring). Furthermore to properly use the USB you must remove the pull-up resistor used for the I2C bus.

- **Thermal sensor**
  The DS1820 or the LM35 can be used. The DS1820 requires the pull-up resistor while the LM35 does not. Refer to the Thermal sensor paragraph for more information.

| Components | |
|---|---|
| **R1 =** 2.2KΩ 1/4W 5% | **C14 =** 22µF 10V electrolytic |
| **R2 =** 2.2KΩ 1/4W 5% | **C15 =** 0.1µF 50V polyester |
| **R3 =** 10KΩ 1/4W 5% | **C16 =** 0.1µF 50V polyester |
| **R4 =** 4.7KΩ 1/4W 5% | **C17 =** 0.1µF 50V polyester |
| **R5 =** 330Ω 1/4W 5% | **C18 =** 0.1µF 50V polyester |
| **R6 =** read “Transmission Protocols” p. | **C19 =** 0.1µF 50V polyester |
| **R7 =** read “Transmission Protocols” p. | **C20 =** 0.1µF 50V polyester |
| **R8 =** 100KΩ 1/4W 5% | **C21 =** 0.1µF 50V polyester |
| **R9 =** 10KΩ 1/4W 5% | |
| **R10 =** 4.7KΩ 1/4W 5% | **D1 =** 1N4148 |
| **R11 =** 220Ω 1/4W 5% | **D2 =** 1N4004 |
| **R12 =** 220Ω 1/4W 5% | **D3 =** 1N4004 |
| **R13 =** 220Ω 1/4W 5% | **D4 =** 1N4004 |
| **R14 =** 220Ω 1/4W 5% | **D5 =** 1N4004 |
| **R15 =** 220Ω 1/4W 5% | **D6 =** 1N4004 |
| **R16 =** 220Ω 1/4W 5% | **LED1 =** LED red 3mm |
| **R17 =** 220Ω 1/4W 5% | |
| **R18 =** 220Ω 1/4W 5% | **IC1 =** 40 pins PIC |
| **R19 =** 220Ω 1/4W 5% | **IC2 =** MAX481CPA |
| | **IC3 =** MAX232 |
| **C1 =** 15pF ceramic | **IC4 =** PCF8563P |
| **C2 =** 15pF ceramic | **IC5 =** 24LC128P |
| **C3 =** 0.1µF 50V polyester | **IC6 =** 7805 |
| **C4 =** 0.1µF 50V polyester | **IC7 =** 78xx |
| **C5 =** 0.1µF 50V polyester | |

| | |
|---|---|
| **C6 =** 0.1µF 50V polyester | **DS1820 =** DS1820 or LM35 (read notes) |
| **C7 =** 0.1µF 50V polyester | **Q1 =** Quartz 32.768Hz |
| **C8 =** 0.1µF 50V polyester | **Q2 =** Quartz 20MHz |
| **C9 =** 1µF 50V electrolytic | **FR =** photo resistor 100KΩ |
| **C10 =** 1µF 50V electrolytic | |
| **C11 =** 1µF 50V electrolytic | **PORTA** = ML10E |
| **C12 =** 1µF 50V electrolytic | **PORTB** = ML10E |
| **C13 =** 100µF 25V electrolytic | **PORTC** = ML10E |
| **PORTD** = ML10E | **SV9** = I2C connector |
| **PORTE** = ML10E | **X1 =** female DB9 |
| **PROG =** ML10E | |
| **LCD** = LCD display | **BATTERY =** Battery connector |
| | **RS232 =** read “Transmission Protocols” p. |
| **POWER =** Power connector | **RS485** = read “Transmission Protocols” p. |
| **RESET =** micro switch (button) | **RX =** read “Transmission Protocols” paragraph |
| **SPK =** 5V buzzer | **TX =** read “Transmission Protocols” paragraph |
| | **SERVO =** Servo motor power connector |
| **SV1 =** Servo motor connector | **ECHO =** read “Ultrasonic sensor” paragraph |
| **SV2 =** Servo motor connector | **TRIG =** read “Ultrasonic sensor” paragraph |
| **SV3 =** Servo motor connector | |
| **SV4 =** Servo motor connector | |
| **SV5 =** Servo motor connector | |
| **SV6 =** Servo motor connector | |
| **SV7 =** Servo motor connector | |
| **SV8 =** Servo motor connector | |

## Assembling instructions

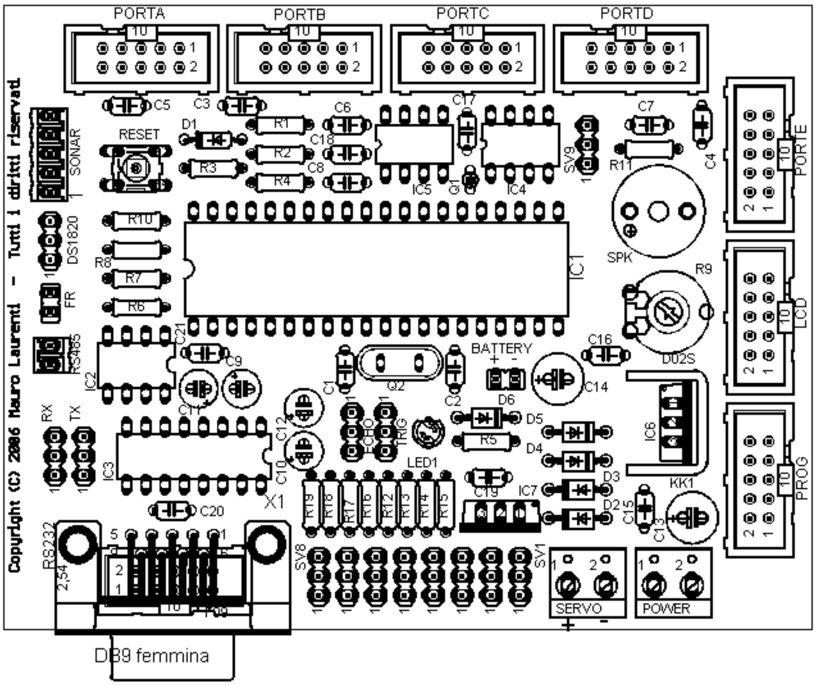



**Figure 19:** *Freedom after the assembling*

Freedom is an embedded system made with a double layer PCB and through hole components. The PCB can be requested using the service provided by LaurTec www.LaurTec.com.
The cost is a free donation used to support the free projects. Its assembly is not very difficult since no SMD components are used, but some attentions are required anyway. To simplify the assembly, the PCB has a component lithography with names. Figure 19 shows the assembled PCB. For the assembly is useful to follow the rule of higthness, mounting the low profile components first.

The resistors will be mounted first than the diodes, the resistors must be mounted according to the color coding while the diodes must be mounted according to the cathode and anode as shown in Figure 19. The diodes have a white or black ring that points the cathode. This ring is also shown on the PCB lithography[26].

After the low profile components assembly, it is possible to proceed mounting the by-pass capacitors of 0.1uF and the ceramic capacitors. This order is not mandatory but it could be handy.

The ICs should be mounted on IC sockets, this will avoid damaging the IC during the assembling and keeping the replacement easier[27]. This is also handy considering that you can mount different PIC on the board.

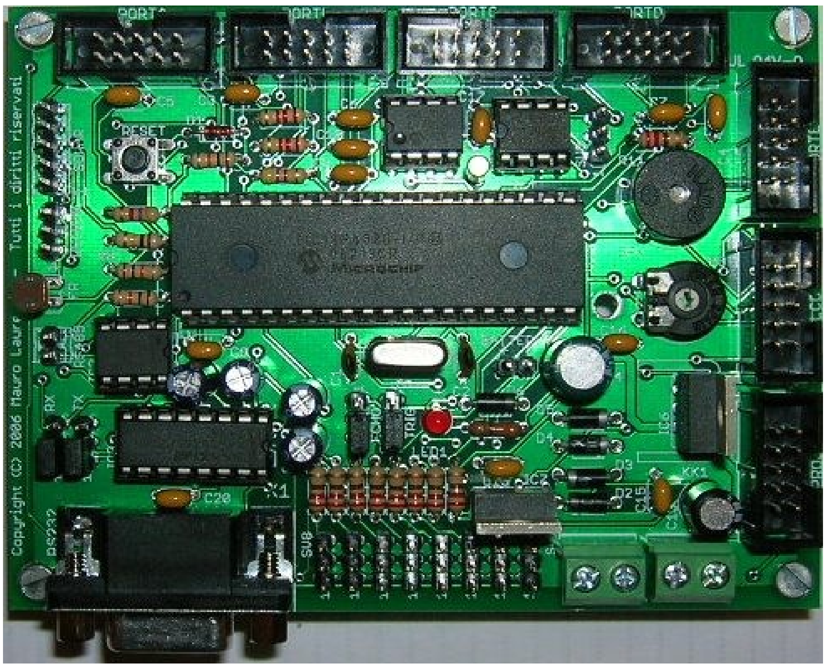


**Figure 20:** *Freedom after the assembling*

[26] LED diodes have the cathode pointed by a truncated case. Another way to figure out who is the cathode is the pin length, the cathode is the short one.

After the IC sockets and the connectors you can mount the electrolytic capacitors, making sure that they are mounted with the right polarity, as shown on the PCB lithography. As shown in Figure 19, the lithography points out the terminal + while on the capacitor cases are normally shown the terminal -.

At the end of the assembly, Freedom will be as shown in Figure 20 or so; in fact some components could be not mounted at all if not required[28].

A little additional note is needed to make the cables required for Freedom. These must be assembled as shown in Figure 21, make sure that the connector guide pin are located as shown in Figure 21 [29].

**Figure 21:** *Freedom external peripherals connection cable*

If you do not have the tool to make the assembly you can use a plier or something similar as large as the connector. You must push the connector keeping the pressure uniform on the connector itself, otherwise you can break it. To ensure a good connection you have to press the cable using the first plastic strip and than apply the plastic latch, on which you don't have to apply any compression.

[27] The IC socket must be soldered first, the IC must be mounted just at the end of the PCB assembly.

[28] The PCB could be changed without any notice and documentation update.

[29] The ribbon cable is the same one used for the PC floppy disk or hard disk drives. The required ribbon cable has only ten wires. This can be purchased or got out of bigger ribbon cables.

## Bibliography

[1] www.LaurTec.com : Electronic web page where you can find the most updated document version and further documents.

[2] www.microchip.com : Microchip technology home page. PIC18F4580 and MCP2551 Datasheet can be downloaded from here.

[a] : DS39500A "PICmicro® 18C MCU Family Reference Manual"

[3] www.maxim-ic.com : Maxim home page.

# History

| Date | Version | Name | Change Description |
|---|---|---|---|
| 25/06/2009 | 1.1 | Mauro Laurenti | Original Translated Version made out of the Italian Version 1.1 |