42 ELECTRONICS

PROBOT

Technical Reference

Authors: Mikel Zabalza, Josep Pinyot, Jaume Garcia

08 03 2018

Version 1.4

CONTENTS

1. INTRODUCTION

Probot is a programable robot, able to make different moves and actions, Its an introduction to robotics and programming, making it more accessible for the youngest audience .This 2-wheel-driven robot is designed to be configured through visual programming.

1.1 MAIN UTILITIES

Perform a series of movements pre introduced by the user through an APP or through a physic panel using specific pieces that represent the movements: go forward, go backward, turn left and turn right. There will be too more specific pieces representing, velocity, performing time for a movement, reproduce sounds and degrees of turn.

Apart from the main utilities our project will be able to **detect colours** from the floor and detect collisions allowing the user create labyrinths himself for solve using the commands and treating the cases of collision and colour error as if the user would have failed, and it could take little metal objects too using an electromagnet.

It could save specific patrons in **memory** for perform complex and varied sets of movements.

The robot contains a **led panel** to be able to show the actual movements that it is performing among other esthetics things as messages, expresions, faces …

1.2 GOAL

The aim of this project is to make a toy in a *computational thinking* concept, So that its users can develop skills such us:

- Develop programs to complete certain tasks (go through a labyrinth, pick up objects…)
- Understand the execution of the selected program
- Learn through error and predict what a program will do.
- Evaluate and refine a simple program.

2. INSTRUCTION MANUAL

Probot is an introduction to robotics and programming for the youngest audience .The robot is designed to execute a set of action through an app, that communicates with the robot via bluetooth, and a grid of different pieces witch you can choose to run specific actions.

The user is able to change between the two possibles inputs with the "mode selection" button, through the colour of the LED the user will know the selected input (blue for bluetooth and red for the grid orders).

2.1 CONFIGURATION WITH THE GRID

The grid is the main part of *Probot*, it is located at the top of the robot to facilitate the interaction. To run *Probot* with the grid make sure the "mode selection" LED is red, if not the user can change from bluetooth to grid mode pressing the button next to it (the "mode selection" button).

Once *Probot* is set in grid mode, the user can place the set of piece he want to move the robot, thanks to the sensor located in front of the robot it will never crash.

The different pieces the user can use are:

GO FORDWARE: to make the robot move forward.

GO BACKWARDS: to make the robot move backwards.

TURN LEFT: to make the robot turn left.

TURN RIGHT: to make the robot turn right.

POWER IMAN: to power the electromagnet located down the robot.

STOP IMAN: to stop the electromagnet located down the robot.

POTENTIOMETER: this piece consist of a simple potentiometer, that the user can place it right after the command pieces to change some values of the action (for FORDWARE and BACKWARDS pieces will change the distance to travel and for TURN pieces will change the angel). Also the user can place one POTENTIOMETER in the first place of the grid to change the speed of all the actions.

GUIDE ME: this pieces is designed to help the youngest play with *Probot,* helped with the distance sensor *Probot* will set automatically the distance and the angle to avoid collision, when it moves and turns respectively.

2.2 CONFIGURATION WITH THE APP

To run *Probot* with the app make sure the "mode selection" LED is blue, if not the user can change from grid to bluetooth mode pressing the button next to it (the "mode selection" button).

The *Probot* app is a easy to use interface with similar options as the grid, the main different with the grid are the limited places were the user can place the pieces in the grid and the memory option.

The memory option is a interesting option when the user find a pattern of pieces that runs the robot without crashing. When the user wants to save a pattern he needs to press the “memory” button on the app, also the user will be able to name the pattern. When the user wants to recover the pattern he will need to press the “recover” button and select the wanted pattern from the menu.

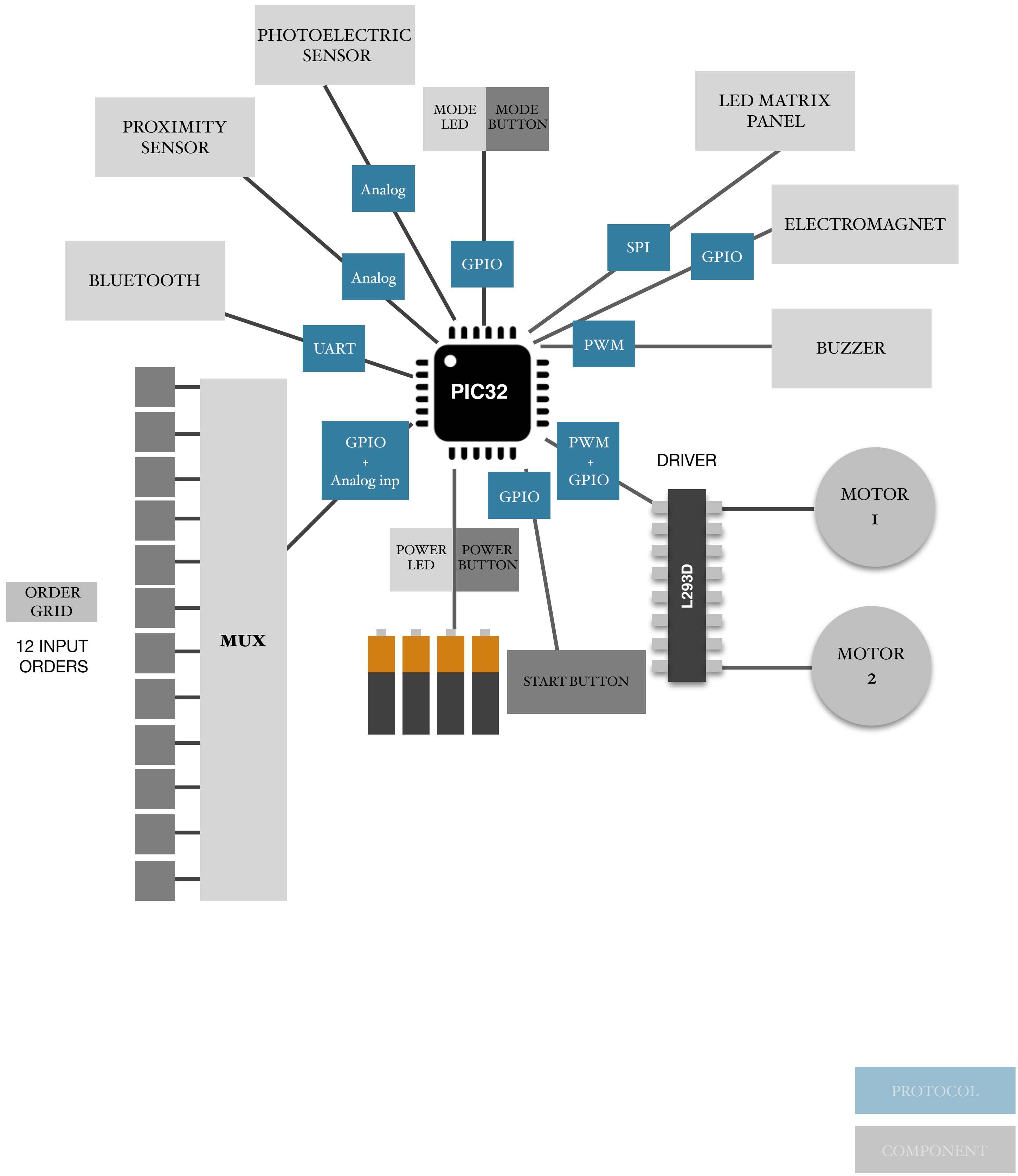
2.3 INTERACTIVE PARTS

Probot will be able to interact directly with the user with the LED panel and the buzzer. Both are build not just to facility and improve the experience with the robot, also thous parts will hep the user to know interesting things about the patterns.

LED PANEL: It will help the user to know the step of his pattern the robot has crashed, showing the number of the piece.

PUZZER: The buzzer will help the user emitting sounds when it crash and a simple music when he found a pattern that runs the *Probot* without crashing.

3. BLOCK DIAGRAM



4. COMPONENTS

4.1 COMPONENT TABLE

Component	Uds	Manufacturer	M. Reference	Supplier	S. Reference	Price (EUR)
Microcontroller	**1**	**Microchip**	**PIC32MX270F256D-50I/PT**	**Farnell**	**2449081**	**4.46**
Proximity sensor	1	SHARP	GP2Y0A41SK0F	Farnell	1618431	15
Female PIN sensor	1	Pololu	3-Pin Female JST PH-style	Exp-tech	-	0.82
Buzzer	1	Pro-signal	ABT-414-RC	Farnell	2098836	1.55
Drive Motor	1	Texas Instruments	L293DNE	Farnell	1470423	1.78
Analog MUX	1	Texas instruments	CD74HC4067SM96	Farnell	1601157	0.4
Led matrix	1	Kingbright	TA08-81GWA	Farnell	2290396	6.15
Shift Register	2	Texas Instruments	CD74HC164M	Farnell	1103161	0.589 x 2 = 1.17
Bluetooth module	1	MICROCHIP	RN4020-V/RM	Farnell	2442930	9,38
Photoelectric sensor	1	ADVANCED PHOTONIX	NORPS-12	Farnell	327700	3.18
Potentiometer	1	BOURNS	3362P-1-103LF	Farnell	9354391	1
Mode Switch	2	NKK switches	SS12SDP2	Farnell	1524325	0.05 x 2 = 0.1

4.1 MICROCONTROLLER

PIC32MX270F256D-50I/PT

Fabr. Ref	PIC32MX270F256D-50I/PT
Farnell Code	2449081
Frecuency:	50MHz
Pins:	44 where 34 are I/O.
Interfaces:	UART, SPI, I²C, I²S, USB
RAM:	64KB
Memory size:	256KB
Consume (50 Mhz)	25mA
USED PINS	

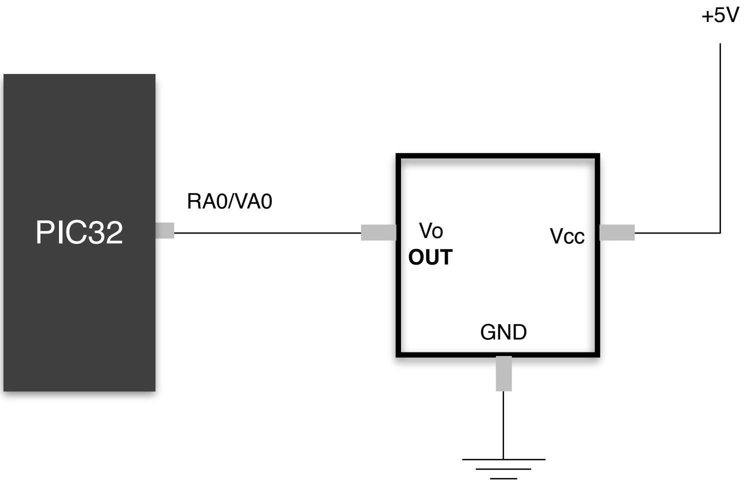
USED PINS:

CONNECTED COMPONENT	PIN	FUNCTION
POWER	1	RB9 – I
MODE	2, 3, 4	RC6 – I/ RC7 – O/ RC8 – I
GRID ON	5	RC9 – O
PROXIMITY SENSOR	11	AN11 – I
GRID	36, 30, 31, 32, 33, 34	AN12 — I / RA2 — O / RA3 — O / RA8 — O / RB4 — O / RA4 — O
DECODER GRID	8, 9	RB10 – O / RB11 – O
MTR DRIV PULSE	19, 20	RPA0 – CO / RPA1 – CO
MTR DRIV MTR1	21, 22	RB0 – O/ RB1 – O
MTR DRIV MTR2	23, 24	RB2 – O/ RB3 – O
ELECTROMAGNET	27	RC2 — O
MATRIX LED	14, 41, 43, 44	SCK / RPB5 — SPI / RPB7 — SPI / RPB8 — SPI
BUZZER	15	RPB15 – CO
BLUETOOTH	12, 13, 25, 26	RA10 – O / RA7 – O / RPC0 – Rx / RPC1 – Tx

I = Input O = Output CO = Compare Output

4.2 PROXIMITY SENSOR

The infrared proximity sensor will detect how far away an object is, the PIC32 micro controller will interpret the input and generate an order. Example: stop the motors before hitting an obstacle.



The sensor will output an analog voltage depending how far away an object is from the sensor. It will be connected to the micro controller using the Analog to Digital converter pin.

SNS-US020 Ultrasonic Sensor

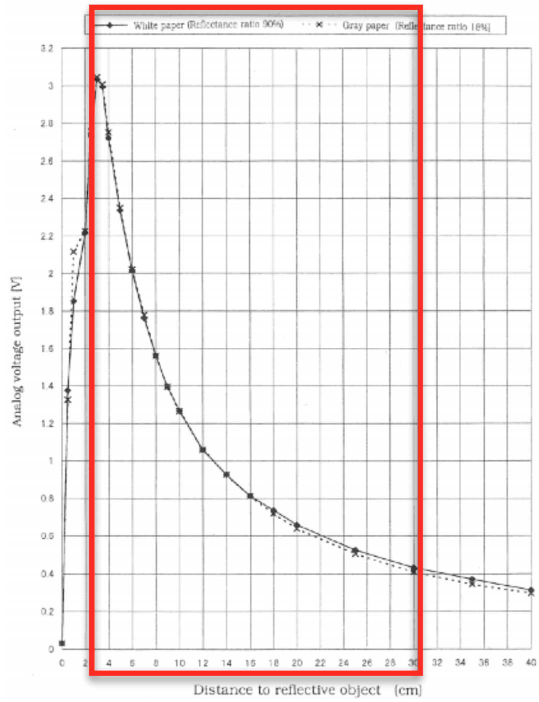
- **Manufacturer** SHARP
- **Supplier** Farnell
- **Manufacturer Reference** GP2Y0A41SK0F
- **Supplier Reference** 1618431
- **Range** 4 to 30cm
- **Power supply** 4.5 to 5.5 V
- **Analog voltage output** 0.4V
- **Consumption** 30 mA
- **Price** 7.8 EUR

The sensor needs a 3 pin adapter to be able to connect it to the PCB.
The 3-Pin Female JST PH-Style Cable has been selected for this purpose

3-Pin Female JST PH-Style Cable

- **Manufacturer** Pololu
- **Supplier** exp-tech
- **Price** 0.82 EUR

The graphic below taken from the data sheet shows how the voltage output from the sensors varies with the distance detected. With distances below 4cm we could get repeated voltages and above 30 cm, they are not conclusive.
Considering this we will able to assign the different voltage inputs between 0, 4 and 3 V to its corresponding distance, in our PIC32. However we will set 0 and 5V as reference voltages. The digital value is calculated based on following equation.

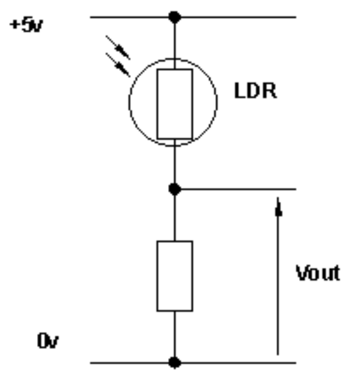


Interfacing to the microcontrollers is straightforward. The sensor will output a single analog output voltage (OUT) connected to an analog-to-digital converter for taking distance measurements. The 10-bit ADC pin of the PIC provides readings from 0 to 1023 as the input voltage rises from 0 to Vdd (5V).
The sensor automatically updates the output approximately every 16 ms.

4.3 PHOTOELECTRIC SENSOR

The LDR will be used with a RGB LED to detect colours. Measuring the LED's light reflected in the surface we want measure having like that 3 measures one for RED, another for GREEN and one last for BLUE, with those three measures we can determinate a colour.

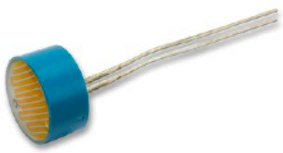
For the LDR one side of the sensor will be connected to the power supply while the other will be connected to an analogical input pin of the microcontroller. The analog input will be connected to a pull-down resistor to ground.



we will use the model LDR CdS PHOTO CELL from Farnell with the specifications:

LDR CdS PHOTO CELL

- **Manufacturer** ADVANCED PHOTONIX
- **Supplier** Farnell
- **Supplier Reference** NORPS'12
- **Supplier Reference** 327700
- **Darkness resistance** 1 Mohm
- **Darkness resistance** 100 Kohm

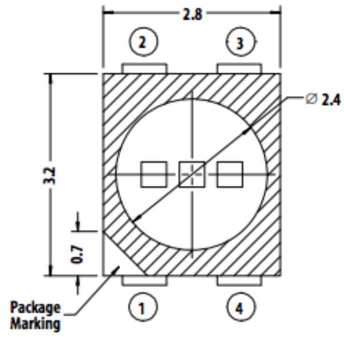


RGB LED

- **Manufacturer** BROADCOM LIMITED
- **Supplier** Farnell
- **Supplier Reference** ASMB-MTB0-0A3A2
- **Supplier Reference** 2401105
- **Direct Tension** R 2.1V / G 3.1V / B 3.1V
- **Direct Current** R 20mA / G 20mA / B 20mA

For the RGB LED we have 4 pins:

Lead Configuration	
1	Common Anode
2	Cathode (Blue)
3	Cathode (Green)
4	Cathode (Red)



The anode will be connected to GND and the others connected to a resistor and a digital pin.

4.4 ORDER GRID

As explained in the instruction manual, the robot could be programmed plugging some pieces directly into the car. So the car is going to have 16 holes available for 10 different types of pieces(each piece representing an order, such us, go straight/backwards, turn left-right, activate magnet…). A 16-Channel Analog Multiplexer has been selected to connect 16 inputs of the order grid with the micro controller.

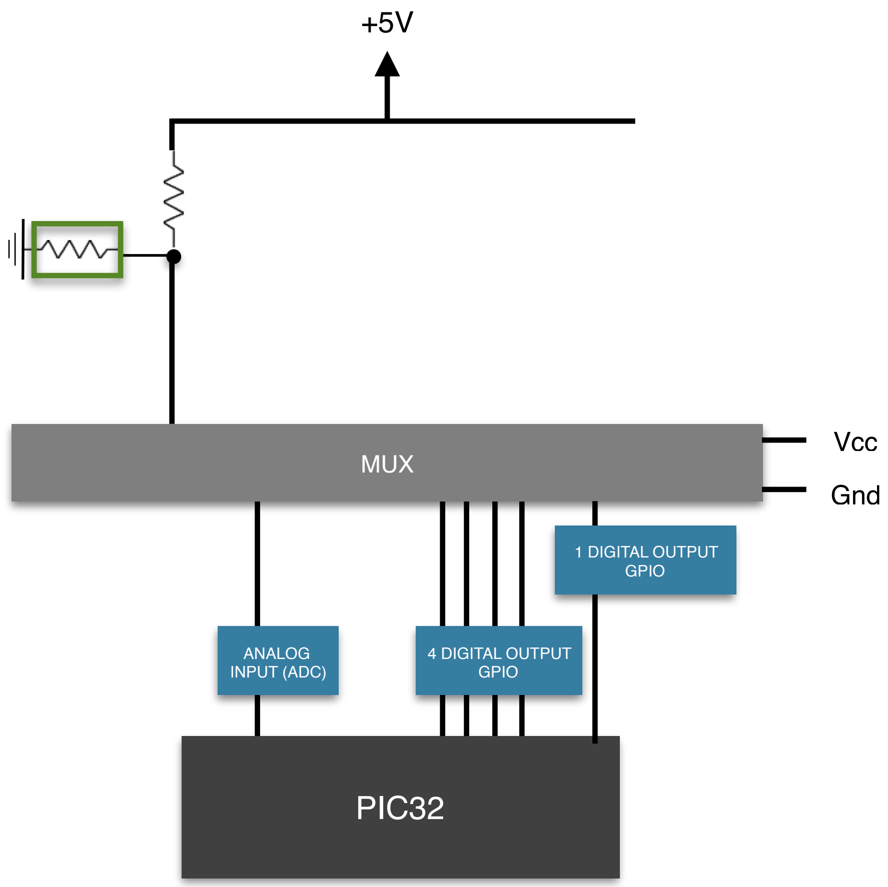
The multiplexer is a combinational circuit device that selects one of N inputs and provides it on its output. We can determine which input should be passed to the output trough a number of digital inputs called select lines. If there are n select lines, then the maximum input lines are 2^n.
In this case, with a multiplexer of 16 analog inputs we need 4 select lines. The 4 select lines will be General-purpose input/output (GPIO), configured as digital outputs from our PIC32. The figure below shows how the combination of all 4 select lines will allow us to select the desired analog output (L= Low, H = High):

S1 INPUT	S2 INPUT	S3 INPUT	S4 INPUT	E ENABLE	OUTPUT
L	L	L	L	L	D0
L	L	L	H	L	D1
L	L	H	L	L	D2
L	L	H	H	L	D3
L	H	L	L	L	D4
L	H	L	H	L	D5
L	H	H	L	L	D6
L	H	H	H	L	D7
H	L	L	L	L	D8
H	L	L	H	L	D9
H	L	H	L	L	D10
H	L	H	H	L	D11
H	H	L	L	L	D12
H	H	L	H	L	D13
H	H	H	L	L	D14
H	H	H	H	L	D15

The selected multiplexer has also an ENABLE input, which has to to be activated to perform the multiplexer operation. This will be also an general purpose input/output (GPIO) from the PIC32. This integrated circuit has active LOW ENABLE and the 4 select lines as input and gives complementary outputs.

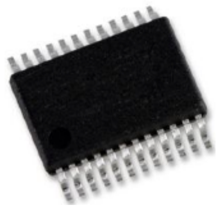
The image below shows the scheme of the input grid - mix - PIC32. The order pieces are represented with the green boxes.

un



This is the selected MUX:

16 :1 Analog Multiplexer



- **Manufacturer** TEXAS INSTRUMENTS
- **Supplier** Farnell
- **Manufacturer Reference** CD74HC4067SM96
- **Supplier Reference** 1601157
- **Power supply** 2 to 6 V
- **PINS** 24

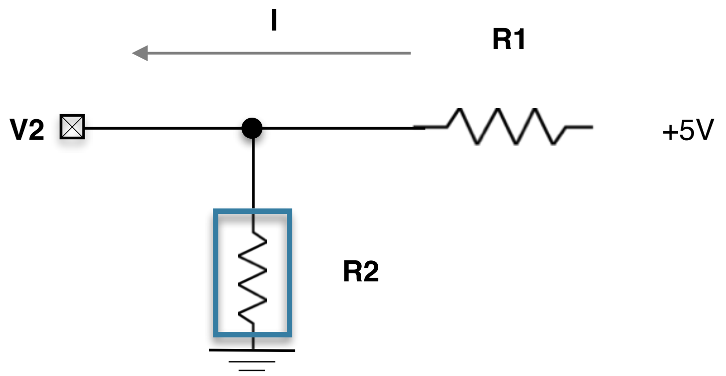
The Analog input received by the PIC32 will be a voltage value between 0 and 3.3 V. The PIC receive this input through its 10 - bit analog to digital conversion input (ADC). This pin is able to read 1024 different values between voltage 0 and 3.3 V. In this project there will be just 10 different cases, representing 10 different inputs (10 different R2).

Selecting a value of R1 (50 Ohm) and combining different input values of R2, we obtain different voltage outputs, as it is shown in the table bellow.

V = R x I
R = R1 + R2

I = V/(R1 + R2)
I = 5/(50 + R2)

V2 = V1 - (R1 x I)
V2 = 5 - (R1 x I)

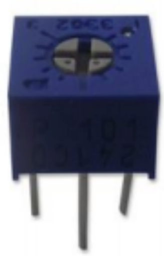


V1(V)	5		
R1(Ohm)	10000		
	R2 order x **(Ohm)**	**I (A)**	**V2 (V)**
R2 Potentiometer	700	0,00047	0,33
R2 order 2	1500	0,00043	0,65
R2 order 3	2800	0,00039	1,09
R2 order 4	4000	0,00036	1,43
R2 order 5	5500	0,00032	1,77
R2 order 6	7000	0,00029	2,06
R2 order 7	9000	0,00026	2,37
R2 order 8	12000	0,00023	2,73
R2 order 9	15000	0,00020	3,00
R2 order 10	20000	0,00017	3,33

The first order will be reserved for the potentiometerter order piece. For this, we will use a 10 Ohm potentiometer which will allow us to reserve the range between 0 and 0,83 V for this utility.

3362P-1-103LF Potentiometer

- Fabricant Bourns
- Fabricant Ref 3362P-1-103LF
- Supplier Farnell
- Supplier Ref 9354301
- Resitance 10 Ohms
- Clock Frecuence 30Mhz



4.5 MOTOR

The motor is the output element, responsible of the robots 4 basic movements: Forward, Backward, left turning, right turning. Two motors respectively connected to independent wheels, will be enough for this. This are the main characteristics of the selected motors:

MOTOR MG-6-120

- **Supplier** Olimex
- **Supplier Reference** MG-6-120
- Double shaft Gear Motor
- **Voltage** 3 - 6 VDC
- **Current** < 300 mA
- **Speed** 100 rpm
- **Torque** 1.0 kgf.cm
- **Price** 2.5 EUR

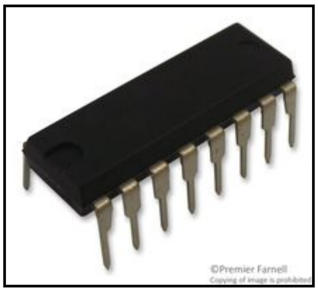
The motors can not be controlled directly from the micro controller pin. Thus we need a motor controller.

4.5.1 MOTOR DRIVER

The motors will be controlled through the motor driver **L293D:**

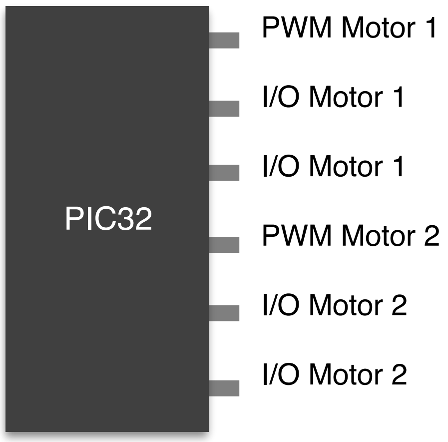
MOTOR DRIVER L

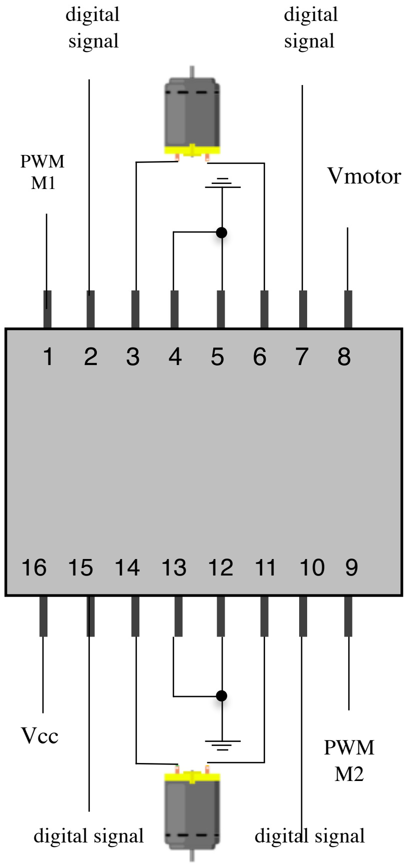
- **Manufacturer** Texas Instruments
- **Manufacturer Reference** L293DNE
- **Supplier** Farnell
- **Supplier Reference** 1470423
- **Supply Voltage min/max (Motor Voltage)** 4.6/36V
- **Output Voltage** 39V
- **Maximum Continuous Motor Current** 600mA
- **Output Current** 1.2A
- **Nº Pins** 16
- **Price** 1.784 EUR



The L293D is a 16-Pin Motor Driver IC, capable of running two DC motors at any direction at the same time. The Driver works on the principle of **Half h-Bridge,** which is a set up used to run motors both in clock and anti clockwise direction. A PWM signal from the PIC will tell the power transistor when to turn the motor on and when to turn the motor off. The PWM's duty cycle will determine the speed the motor turns.

This schema together with the table below, summarizes the connection between the PIC, the motor Driver, and the two motors





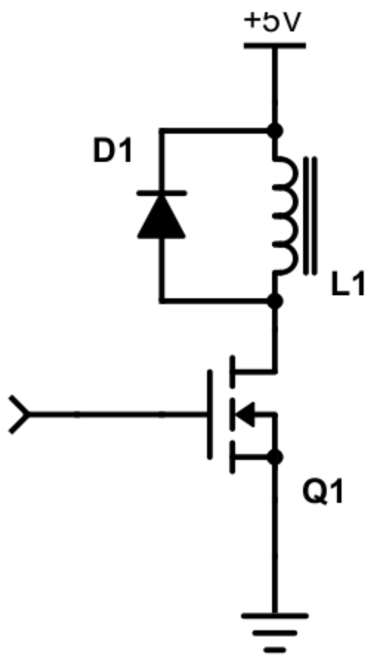
PIN	DESCRIPTION	PIN	DESCRIPTION
1	Control of Voltage supplied to the motor. PWM signal	9	Control of Voltage supplied to the motor. PWM signal
2	Control of the direction of the motor rotation	10	Control of the direction of the motor rotation
3	Motor 1	11	Motor 2
4	Ground pins connected to ground of circuit (0V)	12	Ground pins connected to ground of circuit (0V)
5	Ground pins connected to ground of circuit (0V)	13	Ground pins connected to ground of circuit (0V)
6	Motor 1	14	Motor 2
7	Control of the direction of the motor rotation	15	Control of the direction of the motor rotation
8	Motor Voltage supply	16	Circuit Voltage supply (5V)

PWM *Pulse-Width Modulation*
I/O *Digital Input/Output*

4.6 ELECTROMAGNET

To drive the electromagnet we will need more than 3V, for that reason we will use one Mosfet transistor with the gate connected to the PIC32, the source connected to ground and the drain connected to the ground connection of the electromagnet. The electromagnet will go connected to the battery directly.

As we didn't found a electromagnet that meets our requirements, we will buy a relay to disassemble and collect the coil to use it our project as our electromagnet.



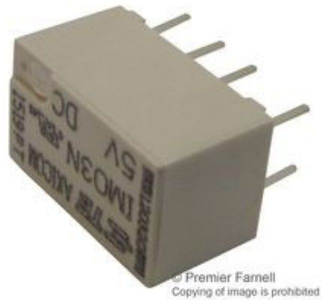
Mosfet transistors 2N7002

- **Manufacturer** Microchip
- **Manufacturer Reference** 2N7002-G
- **Supplier** Farnell
- **Supplier Reference** 2775064
- **Transistor polarity** N-channel
- **Price** 0.175 EUR

Standard Recovery Diode

- **Manufacturer** Multicomp
- **Manufacturer Reference** 1N4001
- **Supplier** Farnell
- **Supplier Reference** 9564993
- **Repetitive Reverse Voltage** 50V
- **Forward Voltage VF Max** 1.1V
- **Price** 0.07 EUR

Signal relay



- **Manufacturer** AXICOM
- **Manufacturer Reference** IM03NS
- **Supplier** Farnell
- **Supplier Reference** 1770584
- **Coil Voltage** 5V
- **Price** 1.66 EUR

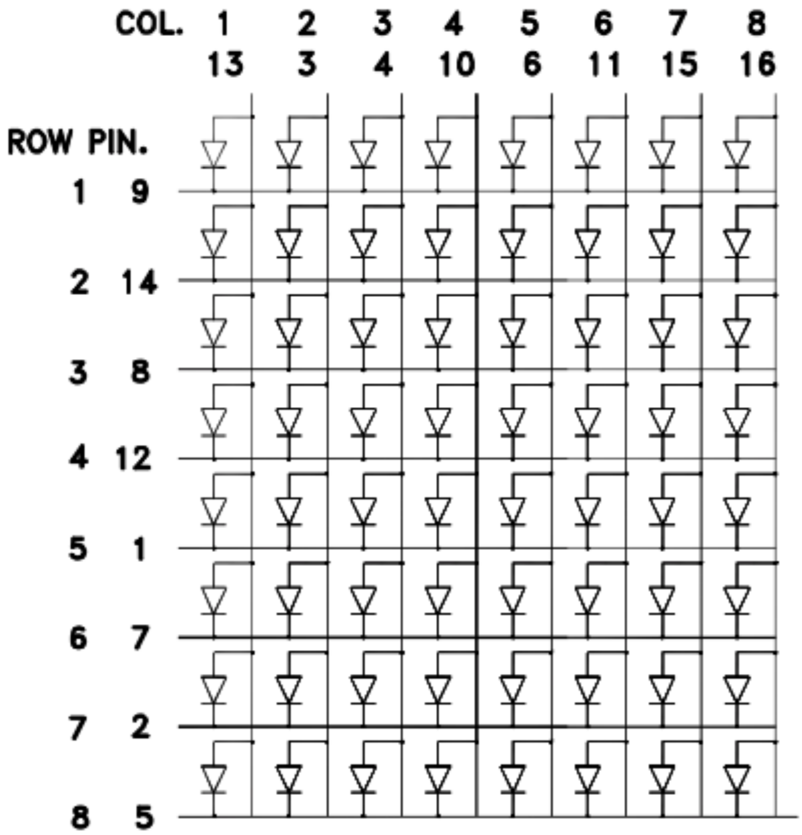
PROTOCOL	DESCRIPTION
GPIO	This port will be connected to the transistor in order to power the electromagnet.

4.8 LED MATRIX

Our LED matrix works on the principle of multiplexing. The PIC32 is connected to the shift register and sends the data and the clock by different lines .

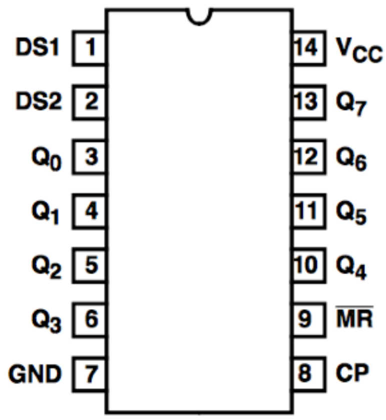
LED Dot Matrix

- **Manufacturer** Kingbright
- **Manufacturer Reference** TA08-81GWA
- **Supplier** Farnell
- **Supplier Reference** 2290396
- **LED color** Green
- **Supply Voltage** 2.2V
- **Display Dots x Lines** 8x8
- **Price** 6.15 EUR

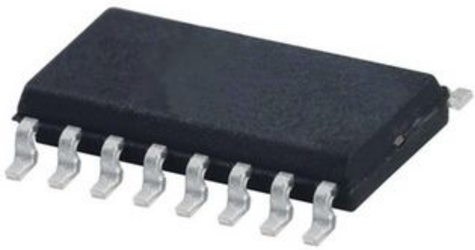


4.8.1 SHIFT REGISTER

To power our LED matrix we will use two 8-bits shift registers (serial-in, parael-out), connecting the outputs of one of them direct to the rows(positive), and the outputs of the other connected to mosfet transistor(with source connected to ground and drain to the column of the LED).



Shift Register 74HC164



- **Manufacturer** Texas Instruments
- **Manufacturer Reference** CD74HC164M
- **Supplier** Farnell
- **Supplier Reference** 1103161
- **Supply Voltage min/max** 2/6V
- **Nº Pins** 14
- **Nº of Bits Per Element** 8bit
- **Price** 0.58 EUR

Mosfet transistors 2N7002

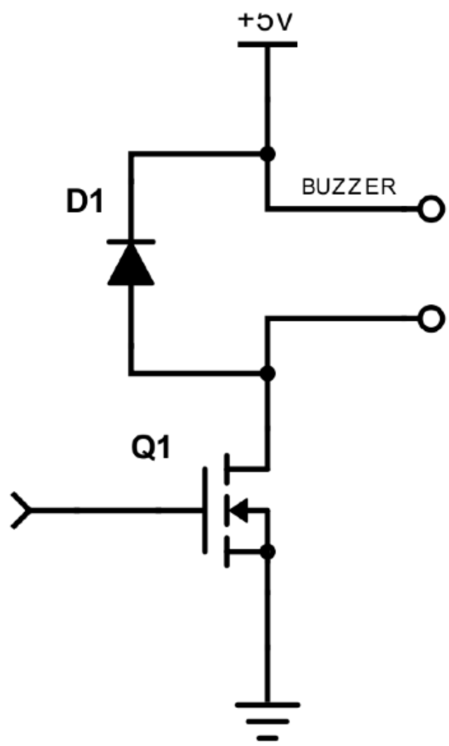
- **Manufacturer** Microchip
- **Manufacturer Reference** 2N7002-G
- **Supplier** Farnell
- **Supplier Reference** 2775064
- **Transistor polarity** N-channel
- **Price** 0.175 EUR

4.9 BUZZER

The buzzer is one of our output elements build to ease the interaction with the device. To control the buzzer we will connect the output from the PIC32 to the gate of one Mosfet transistor, with source connected to ground and the drain connected to the output of the buzzer. In this way we will be able to run the buzzer with more voltage than using the output of the PIC32.

To prevent the PIC32 and the transistor of suffering back-emf, we will put a suppression diode across the buzzer.

To generate an audio signal to play on the buzzer we will set the PWM period to 1/frequency of the desired sound, So we can generate different frequencies.



Buzzer

- **Manufacturer** Pro-signal
- **Manufacturer Reference** ABT-414-RC
- **Supplier** Farnell
- **Supplier Reference** 2098836
- **Supply Voltage Min/Max** 3/8V
- **Sound Level SPL** 85dB
- **Price** 1.55 EUR

Mosfet transistors 2N7002

- **Manufacturer** Microchip
- **Manufacturer Reference** 2N7002-G
- **Supplier** Farnell
- **Supplier Reference** 2775064
- **Transistor polarity** N-channel
- **Price** 0.175 EUR

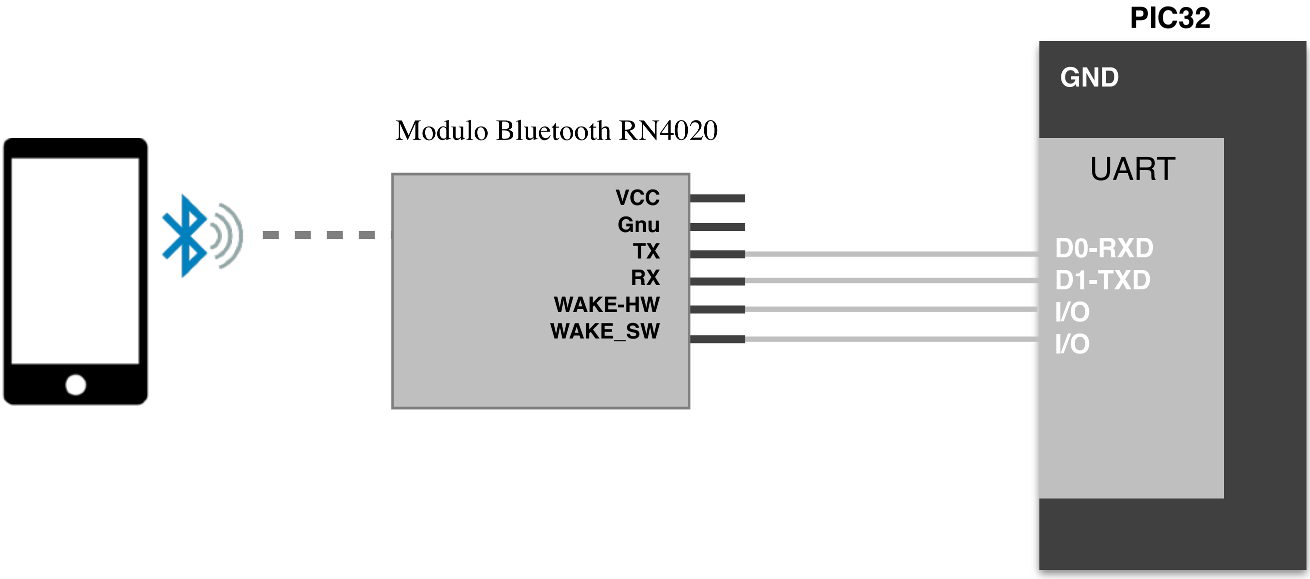
Standard Recovery Diode

- **Manufacturer** Multicomp
- **Manufacturer Reference** 1N4001
- **Supplier** Farnell
- **Supplier Reference** 9564993
- **Repetitive Reverse Voltage** 50V
- **Forward Voltage VF Max** 1.1V
- **Price** 0.07 EUR

PROTOCOL	DESCRIPTION
PWM	This output will be connected to the transistor in order to control the frequency we want to reproduce, but also to turn on and off the sound.

4.10 BLUETOOTH

For our bluetooth communication we have chosen a low energy **Bluetooth module RN4020** by Microchip. Using this module we can receive and send information from our PIC to a mobile application or a computer. The communication between PIC and RN4020 are established through one of the UART modules present in our PIC.

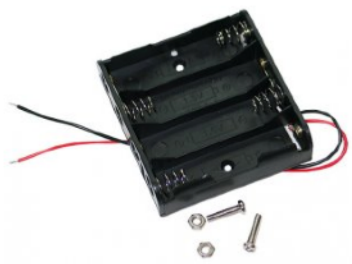


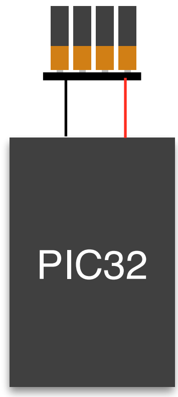
The WAKE_HW and WAKE_SW pins of the bluetooth module control the sleep, activation and reset modes of the module. The c

BLUETOOTH MODULE RN4020

- **Operating Voltage** 3.3V
- **Input Power Supply** 3.6V - 6V
- **Frequency Band** 2.4GHz 2. 48GHz
- **Internal flash memory** 64KB
- **Price** 9.38

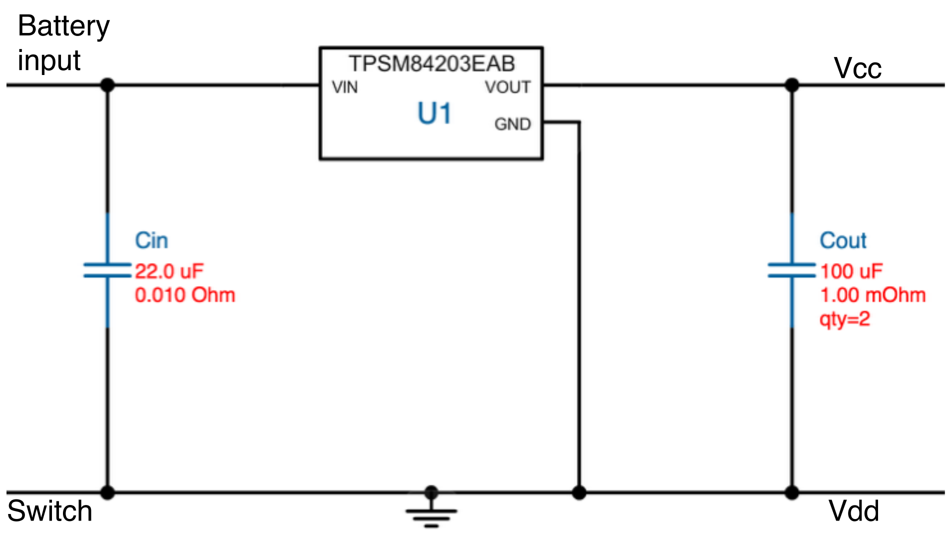
4.11 BATTERY





The voltage will be supplied to the circuit through four A4 batteries connected in series. This way 5V will be supplied using rechargeable A4 batteries.
However, as our PIC32 operates in **3.3V** . A voltage regulator will be added between the battery and the PIC voltage input to get the desired voltage.

The output of the battery will directly connected to a switch witch allow to have a direct control to to entire circuit and this one will be connected to to the voltage regulator
We will use a TPSM84203EAB voltage regulator following the next connection schema:



Voltage regulator

- **Manufacturer** TEXAS INSTRUMENTS
- **Manufacturer Reference** LMR14203XMKE/NOPB
- **Supplier** Farnell
- **Supplier Reference** 2323516
- **Input Voltage Min/Max** 4.5V / 42V
- **Output Voltage Min/Max** 0.765V / 34V
- **Output Max Corrent** 0.3A
- **Price** 2.76 EUR

Capacitator Cout

- **Manufacturer** MURATA
- **Manufacturer Reference** GRM31CR61A226KE19L
- **Supplier** Farnell
- **Supplier Reference** 1907515
- **Tension Nominal** 10V
- **Capacitance** 22μF
- **Price** 0.783 EUR
- **Quantity** 2

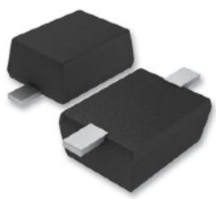
Capacitator Cin

- **Manufacturer** TDK
- **Manufacturer Reference** C1005X6S1C105K050BC
- **Supplier** Farnell
- **Supplier Reference** 2525035
- **Tension Nominal** 16V
- **Capacitance** 1μF
- **Price** 0.136 EUR

Capacitator Cboot

- **Manufacturer** MURATA
- **Manufacturer Reference** GRM188R71A224KA01D
- **Supplier** Farnell
- **Supplier Reference** 1828889
- **Tension Nominal** 16V
- **Capacitance** 0.22μF
- **Price** 0.12 EUR

Diode



- **Manufacturer** DIODES INC.
- **Manufacturer Reference** ZLLS410TA
- **Supplier** Farnell
- **Supplier Reference** 1843764
- **Direct Corrent** 750mA
- **Direct Max Tension** 580mV
- **Price** 0.373 EUR

Inductor

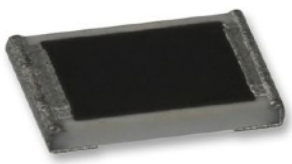
- **Manufacturer** TDK.
- **Manufacturer Reference** CLF6045T-680M
- **Supplier** Farnell
- **Supplier Reference** 2360936
- **Saturation Corrent** 790mA
- **Price** 0.942 EUR

Resistor Rfb2

- **Manufacturer** VISHAY
- **Manufacturer Reference** CRCW04023K32FKED
- **Supplier** Farnell
- **Supplier Reference** 1652792
- **Tension Nominal** 50V
- **Potence Nominal** 63mW
- **Resistance** 3.32kohm
- **Price** 0.0573 EUR

Resistor Rfb1

- **Manufacturer** PANASONIC ELECTRONIC COMPONENTS
- **Manufacturer Reference** ERJ2RKF1021X
- **Supplier** Farnell
- **Supplier Reference** 2302641
- **Tension Nominal** 50V
- **Potence Nominal** 100mW
- **Resistance** 1.02kohm
- **Price** 0.0269 EUR

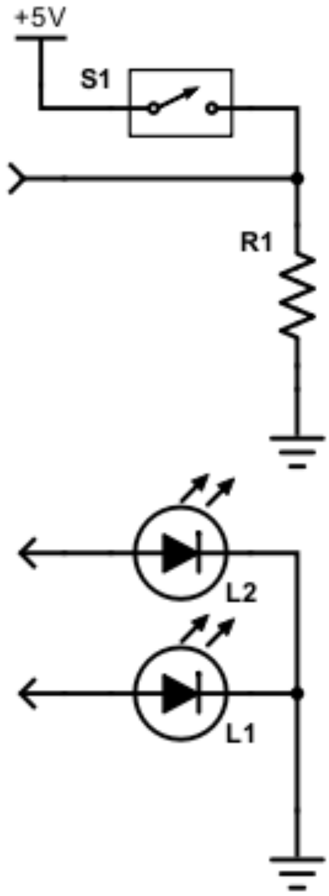


SWITCH

- **Manufacturer** NKKswitches
- **Manufacturer Reference** SS12SDP2
- **Supplier** Farnell
- **Supplier Reference** 1524325
- **Price** 1.63 EUR

4.12 MODE SELECTOR

We decide to implement a simple button to be able to change from grid to bluetooth mode, also we have implement a two colour LED to indicate the mode we are working in.



We will connect the input of the switch to our 5V source, and the output connected to a Pull Down resistance connected to ground. Between the resistance and the switch we will connect our PIC input.
The two LEDs will be connected each one to different outputs of the PIC to indicate in witch mode are the user working in.

LED Blue

- **Manufacturer** Multicomp
- **Manufacturer Reference** MCL053SBLC
- **Supplier** Farnell
- **Supplier Reference** 1581185
- **LED color** Blue
- **Supply Voltage** 3.6V
- **Current** 20mAh
- **Luminous Intensity** 2.500 mcd
- **Price** 0.379 EUR

LED Blue

- **Manufacturer** Rafi
- **Manufacturer Reference** 1.90.690.295/0000
- **Supplier** Farnell
- **Supplier Reference** 9929509
- **LED color** White
- **Supply Voltage** 3.6V
- **Current** 20mAh
- **Luminous Intensity** 2.500 mcd
- **Price** 0.379 EUR

Resistencia SMD de Tipo Chip

- **Manufacturer** Electronics
- **Manufacturer Reference** LR1206-R07FW
- **Supplier** Farnell
- **Supplier Reference** 1100335
- **Resistance** 0.07 ohm
- **Price** 0.546 EUR

SWITCH

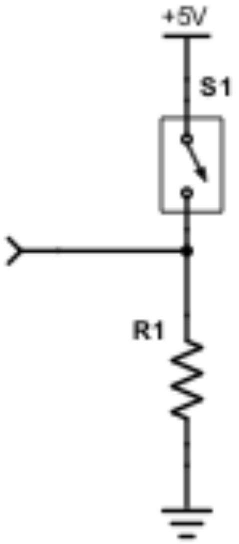
- **Manufacturer** NKKswitches
- **Manufacturer Reference** SS12SDP2
- **Supplier** Farnell
- **Supplier Reference** 1524325
- **Price** 1.63 EUR

PROTOCOL	DESCRIPTION
GPIO IN	This PIC pin will be connected to one of the pins of the switch and it will be connected to the battery. In this way we will use the LOW state of the port to set on mode and the HIGHT to the other mode (grid/bluetooth).
GPIO OUT	It will power one of the LEDs.
GPIO OUT	It will power one of the LEDs.

4.13 START BUTTON

With this button we will be able to send a signal when the user wants the robot to begin to obey the grid orders.

We will connect the input of the switch to our 5V source, and the output connected to a Pull Down resistance connected to ground. Between the resistance and the switch we will connect our PIC input.



Resistencia SMD de Tipo Chip

- **Manufacturer** Electronics
- **Manufacturer Reference** LR1206-R07FW
- **Supplier** Farnell
- **Supplier Reference** 1100335
- **Resistance** 0.07 ohm
- **Price** 0.546 EUR

SWITCH

- **Manufacturer** Multicomp
- **Manufacturer Reference** R13-509A-05-BB
- **Supplier** Farnell
- **Supplier Reference** 1634682
- **Price** 1.42 EUR

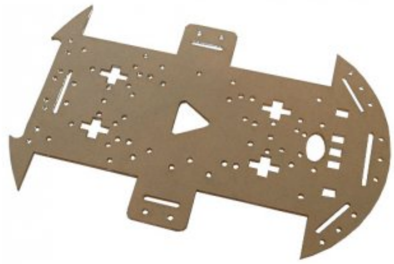
4.14 MAIN ROBOT STRUCTURE

WHEELS

- **Supplier** OLIMEX
- **W1 Supplier Reference** MG-WHEEL
- **W2 Supplier Reference** ROBOT-CW
- **Price** 3 X 1 = 3 EUR

CHASIS

- **Supplier** OLIMEX
- **Supplier Reference** ROBOT-CHASSIS-4
- **Price** 4 EUR



BATTERY HOLDER

- **Supplier** OLIMEX
- **Supplier Reference** BAT-HOLDER-4XAA
- **Price** 0.5 EUR

5 ELECTRICAL CONSUMPTION

With an average use of *Probot* (that means do not push the motors to the maxim power) and using as much devices as possible the calculations for the autonomy would be the next ones:

COMPONENT	AVERAGE CONSUMPTION
Proximity Sensor	12 mAh
Motor	~300 x 2 = 600 mAh
LED Mode	20 mAh
LED RGB	20 mAh
LED Power	20 mAh
LED Matrix (1 LED at a time)	20 mAh
Buzzer	40 mAh
PIC	25 mAh
	757 mAh

With a 900mA battery we will have an autonomy of:

900/757= 1 hora 10 min

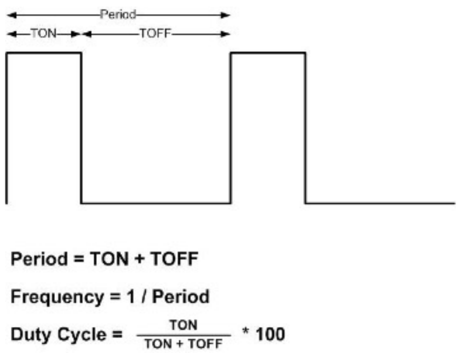
In terms the power:

COMPONENT		AVERAGE POWER
Proximity sensor	12 mA x 5V	0.06W
Motors	300 mA x 2 x 5V	3W
LEDs	20 mA x 4 x 3V	0.24W
Buzzer	40 mA x 5V	0.2W
PIC	25 mA x 3.3	7.57W
		11 W

6 PROTOCOLS

6.1 PWM

The pulse-width modulation (PWM) is a analog protocol that use modulation technique to encode a message into a pulsing signal, this method can be used to transmit information or to control the voltage sent. With PWM we can change the width of the pulse without changing the frequency.



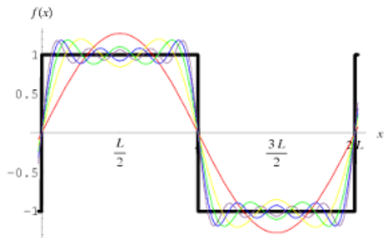
The average value of voltage and current at the output is controlled with the duty cycle, the fraction of one period in which a signal is active, and with the modulation frequency, how many times the pulse went ON and OFF. A low duty cycle corresponds to low power, because the power is off for most of the time.

The main advantage of PWM is the low power loss. When a switch is off there is practically no current, and when it is on and power is being transferred to the load.

BUZZER

We decide to use PWM to be able to control the the output frequency of the buzzer with the frequency of modulation. We will set the PWM period to 1/frequency of the desired sound with a fixed duty cycle(50%,the buzzer will oscillate at its maximum amplitude), to have different sounds with the same volume.

To create a La4 (440 Hz) we will set the duty cycle to 50% and the modulation frequency to 440 Hz. Each second will output 440 pulses of 50% of the pulse ON and 50% of the pulse OFF.



PIC	SHIFT REGISTER	DESCRIPTION
SPI	MR	RESET
SCK	CP	CLOCK
SPI	DS1	DATA
SPI	DS2	ENABLE

6.2 GPIO

General-purpose input/output (GPIO) is a generic pin of one integrated circuit, including whether it is an input or output pin is controllable by the user at run time. It is a rudimentary form of communication where you manually (in code) turn a pin on and off or read its state.

GPIO capabilities may include:

-GPIO pins can be configured to be input or output.
-GPIO pins can be enabled/disabled.
-Input values are readable (typically high or low).
-Output values are writable/readable.
-Input values can often be used as IRQs (typically for wakeup events).

MODE SELECTOR

For this parts we will need three GPIO ports, two outputs and one input. The input will be connected to the switch and will read HIGH or LOW, the two output will be connected to one of the LEDs each.

When input reads LOW, because the switch is off, it will set the robot as GRID mode and will send HIGH to the first output(GRID), and disable the second output(BLUETOOTH). On the contrary when input reads HIGH, because the switch is on, it will set the robot as BLUETOOTH mode and will send HIGH to the second output(BLUETOOTH), and disable the first output(grid).

ELECTROMAGNET

We will set one GPIO pin to output to be able to control the electromagnet, sending HIGH we will enable the electromagnet and we will turn it off disabling the output.

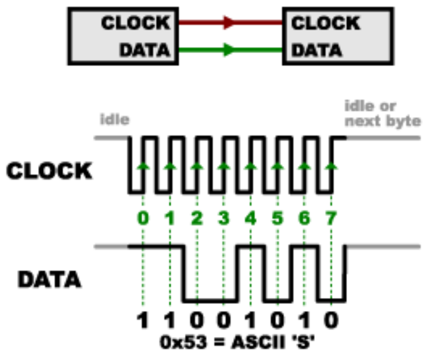
STAR BUTTON

This part have a simple functioning, when the button is pulsed the current flows to a GPIO port set as an input. When the GPIO input reads HIGH the robot will begin to obey the grid orders.

6.3 SPI

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.

SPI is a synchronous protocol, which means that it uses separate lines for data and clock, that keeps both sides in perfect sync. The clock is an oscillating signal that tells the receiver exactly when to sample the bits on the data line, could be the rising or falling edge of the clock signal.



As we can see in the image when the clock pin receive the rising edge checks the state on the data pin. In this we will have the next binary code: 11001010.

Advantages of SPI:

-It's faster than asynchronous serial.
-The receive hardware can be a simple shift register.
-It supports multiple slaves.

Disadvantages of SPI:

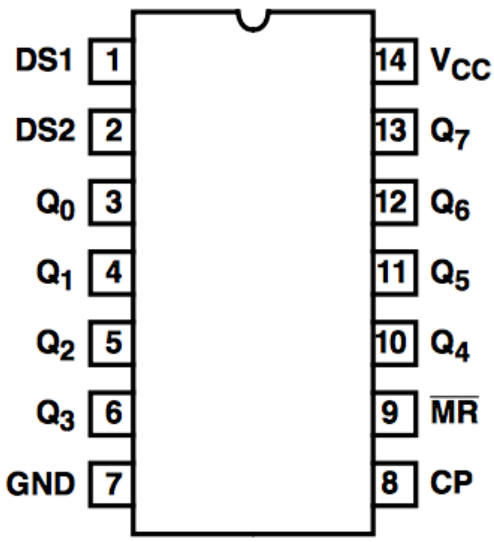
-It requires more signal lines.
-The communications must be well-defined in advance.

LED MATRIX

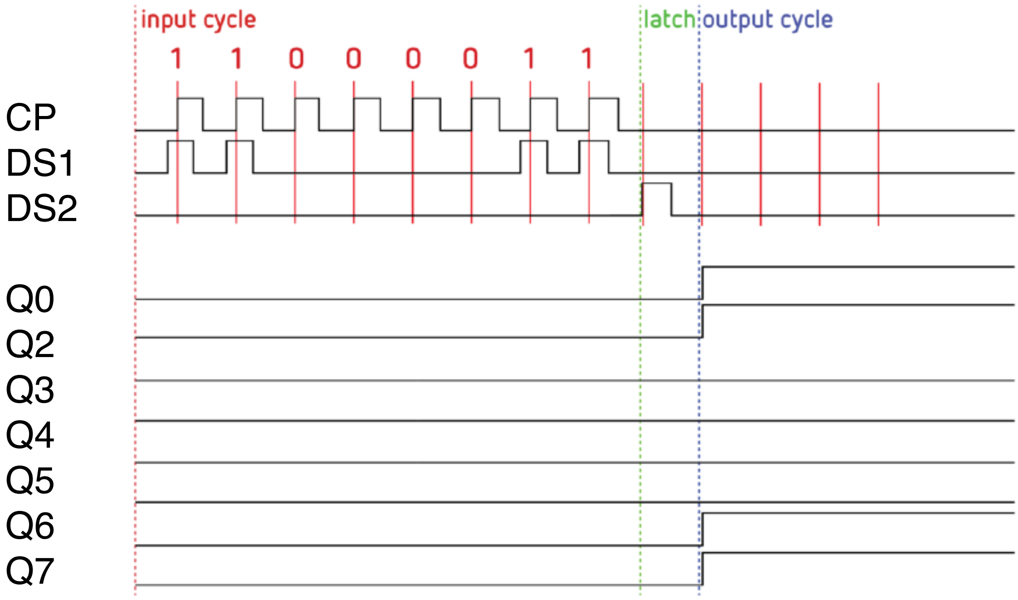
To power our LED matrix we will use two 8-bits shift registers (serial-in, parael-out), connecting the output of one of them direct to the rows(positive), and the outputs of the other connected to mosfet transistor(with source connected to ground and drain to the column of the LED).

We will use this system to enable one column at a time to take advantage of a human eye problem, that maintains the image of the view for about 20 or 30 milliseconds.
With the column shift register we will enable one output at a time, but with the row shift register we will need enable up to the 8 outputs.

Connections:
Because we have two shift registers that need to work together, we will connect the MR, CP and DS2 ports of both shift registers in parallel to the same pins on the PIC.



With the CLOCK input we will receive a regular signal to indicate, in the raising edge, where to read the value in the DATA input.
The DATA input will read a 8 bits(8 raises of the edge of the CLOCK) signal, each of thous bits correspond to a one of the different output. For example to turn on the row number 2, 6, 7 of the column 3 we will send 00100011 to the X edge shift register and 00010000 to the Y edge shift register where 0 is LOW and 1 is HIGH. After receiving the 8 bits signal all the outputs will have memorised their status and sending a HIGH state with the ENABLE input we will enable the selected outputs.
With the RESET input we will have a LOW value when we want to change the memory of the outputs, otherwise the RESET input will recite always a HIGH value.



6.4 UART

We will use UART for implement the Bluetooth mode.

All the information we will send from our APP to UART module will agree this schema.

We will operate under 9600bps, 1 start bit, no parity bit, 1 stop bit and we will use 8 bits whose will be used for save a code to perform the desired actions.

The code we will use to communicate APP and PIC will be the next one:

We will use one bit for determinate wich components we will use. As we have 6 devices we will send as first package 8 bits where bits 0- 5 design usable devices and movement general specs: move forward or backward and turn left or right. And finally two more packages one package specificatin a number from 0 to 360[2] for turn angle and another package specificating a number from 0 to 100[3] representing a percent of velocity.

1ºPackage: General INFO

bit 0 : MOTORS
bit 1 : (IF bit 0) 1 FORWARD 0 BACKWARD
bit 2 : (IF bit 0) TURN LEFT[1]
bit 3 : (IF bit 0) TURN RIGHT[1]
bit 4 : LDR
bit 5 : PROXIMITY SENSOR
bit 6 : ELECTROMAGNET
bit 7 : BUZZER
bit 8 : LED MATRIX

2ºPackage Motor Turn Angle[2]

0	0	0	0	0	0	0	0	0	=	0
1	0	1	1	0	1	0	0	0	=	360

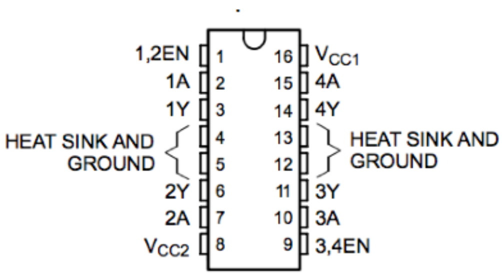
3ºPackage Motor Velocity[3]

0	0	0	0	0	0	0	0	0	=	0%
0	0	1	1	0	0	1	0	0	=	100%

NOTE [1]: if bit 2 and bit 3 are on the behaveur is undefined
NOTE [2]: Specific a number greater than 360 will be in a undefined behaveur
NOTE [3]: The cap will be blocked to 100 for prevent dangerous velocities

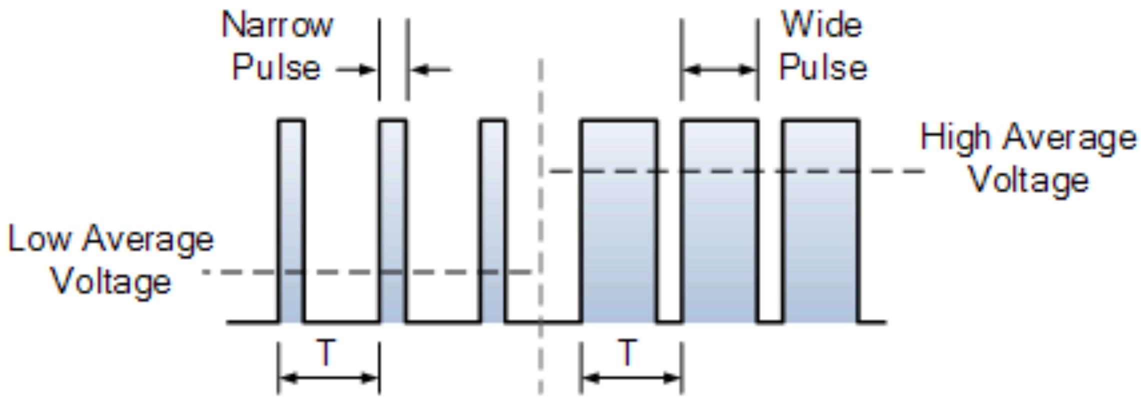
6.5 MOTORS

Give the following tables and schema from the motor driver we will configure the behavior of the motors, using the PWM pulse for increase or decrease the velocity of the motor and for modify the time of functioning choosing the desired angle of rotation.



EN	1A	2A	FUNCTION[1]
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

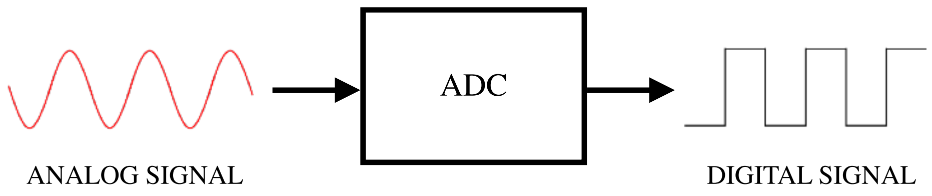
EN	1A	2A	FUNCTION[1]
H	L	H	Turn right
H	H	L	Turn left



A low pulse will produce a slow performance and a high pulse will produce fast performance.

6.6 ANALOG TO DIGITAL CONVERTER (ADC)

Analog to Digital Converter (ADC) is used to convert analog signal-voltage into its equivalent digital number so that the micro controller can process that number.



The main feature of ADC signal is the resolution. The resolution indicates the possible number of digital values that can be converted from an analog input. Our micro controller has in-built 10-bit ADC, this means the digital value received lies between the range 0 to 1023. As the ADC range is 0 - 3.3V we will get a **step size** of 3.3V-2^10 = 3.23 mV.

In the case of proximity sensor we will set different outputs depending the distance to the obstacle. Some values will stop the robot while others will simply reduce its speed. In the case of the input grid we 10 different type of orders, meaning we can get 10 different analog input comping from our multiplexor. Thus the 10 - bit ADC its more than suitable for this purpose.

We can use the triangle method to find the binary representation of an analog input voltage. For example if we get a 2V input from the proximity sensor we will get (1024 x 2V)/5V = 410 V.
The analog input voltage of 2V will be represented by the binary number 110011010.

6.7 PHOTOELECTRIC SENSOR

The LDR will be connected to an analog input pin for measure the voltage of the LDR, meanwhile the LED RGB will be connected to a 3 digital output pins, we will measure the LDR three times, one with RED ON, other with GREEN ON and another with BLUE ON. Due to LDR functioning the voltage will be different with the three inputs and we could measure the RGB intensity in each colour being able to detect colours.