AN-Najah National University

Faculty of Engineering

Department of Computer Engineering



**Project Name**

**Wireless Waiter**

****

Supervisor: **Dr.Ashraf Armosh**

Presented by**: Walaa Abdullah Nawahda**

**Introduction:**

My project is Wireless Waiter which contains two parts: the wireless waiter and the main controller, these two parts communicate between each other by wireless (XBee).I used PIC18F4620 in each of them which I worked on before.

The goal of my project is to get the order for any costumer in the restaurant and meet it by the wireless waiter. On the other hand, make full controlling to the restaurant from getting orders; calculate the account for each costumer.

The wireless waiter is designed to work in different modes: The first mode is (add tables) mode that allows the waiter to add a new table in its memory (EEPROM) which have size of (1024bytes), after enter the number of table the waiter ask to the path of going to and backing from that table. The path could be calculated depends on the number of table or get it by (Wireless). This will do by determine the steps by C# interface; these steps will send from computer to the wireless waiter throw wireless (XBee), the waiter will save the table's number and its path. The second mode is (get order); the waiter will wait to receive the number of costumer's table who want to request, which has been sent by the main controller (throw buttons or by the C# interface), then the waiter will go to it depends on the path which is saved, the waiter will get the order using LCD and Keypad which they exist on waiter's body, then the waiter will back to its initial place. The third mode is (meet order), the waiter will wait to enter the number of order's table, the waiter will go to it depends on the path which is saved, then after the customer take his order there is ability to request additional orders and added to his account , then back to its initial position.

The main controller is designed to organize the orders and the accounts and communicate with the waiter to receive the orders and send the numbers of tables that request a service. First, the main controller connected to button for each table when the costumer chose what he wants, he press the button then the main controller will send the table's number to the waiter and wait the order of the table which is received from the waiter, then display it on computer using C# interface with calculated account.

In this paper, I will discuss how the wireless waiter and the main controller work in details and the main components for each one.

**The Wireless Waiter:**

****

This part is the most problematic part because of controlling motors and the problem of the mechanical base of waiter.

The wireless waiter needs exact movement, especially when meeting the order, if the waiter's base was not work in the way that you want that mean increase the overhead on your code.

Let's look at the waiter's components in details:

**1\_Waiter's Base:**

I had a lot of problems in this part, because of the mechanical work which I don't have any expert in it.

First, I hoped to have a help from outside the university by finding a mechanical expert to make a base as I want, but I didn't find.

After that I tried to find in children's toys to find the base that I want, this way took a lot of times without any useful.

Once, I found one like I want but the problem was the base was weak and can't carry my model and turned just for right direction, so it didn't work well. Until know I don't have solution.

At last, I give up and use a usual car which is not turned exactly with angles, but I didn't find better solution. The advantages of this car are: First, it is strong enough to carry waiter's model. Second; it didn't consume a lot of power (that mean not need to change the battery in short time).The disadvantages of this base is more than its advantages and it limits my works and increase the over head on the code. First, it is not exact in angles which mean the waiter take a lot of space to move between tables. Second, turn the base to right or to the left take more work in the code especially when read the steps of the path which is saved in memory and took throw the wireless.

What I want to do and the waiter's base prevents me:

1\_I wanted to make the waiter move from table to table, but it was so difficult and need more and more space between tables, so I made it to back to its initial position then go to the next table.

2\_I wanted to make a small model for restaurant to show my work but the waiter took larger space than expected. So I couldn’t do it.



**2\_Waiter's Basic Circuit:**

**Components:**

1\_PIC18F4620.

2\_Crystal (4MHZ)

3\_ Capacitors:

12pf (2), 6.8Mf (1), 100nf (3), 1Mf (4),

4\_Resistores:

33K (1), 1K (1)

5\_LED (1N6265)

6\_MAX (232)

7\_CONNECTOR DB9

8\_Bushbuttone (2)

**9\_Regulator Components:**

1\_LM7805C

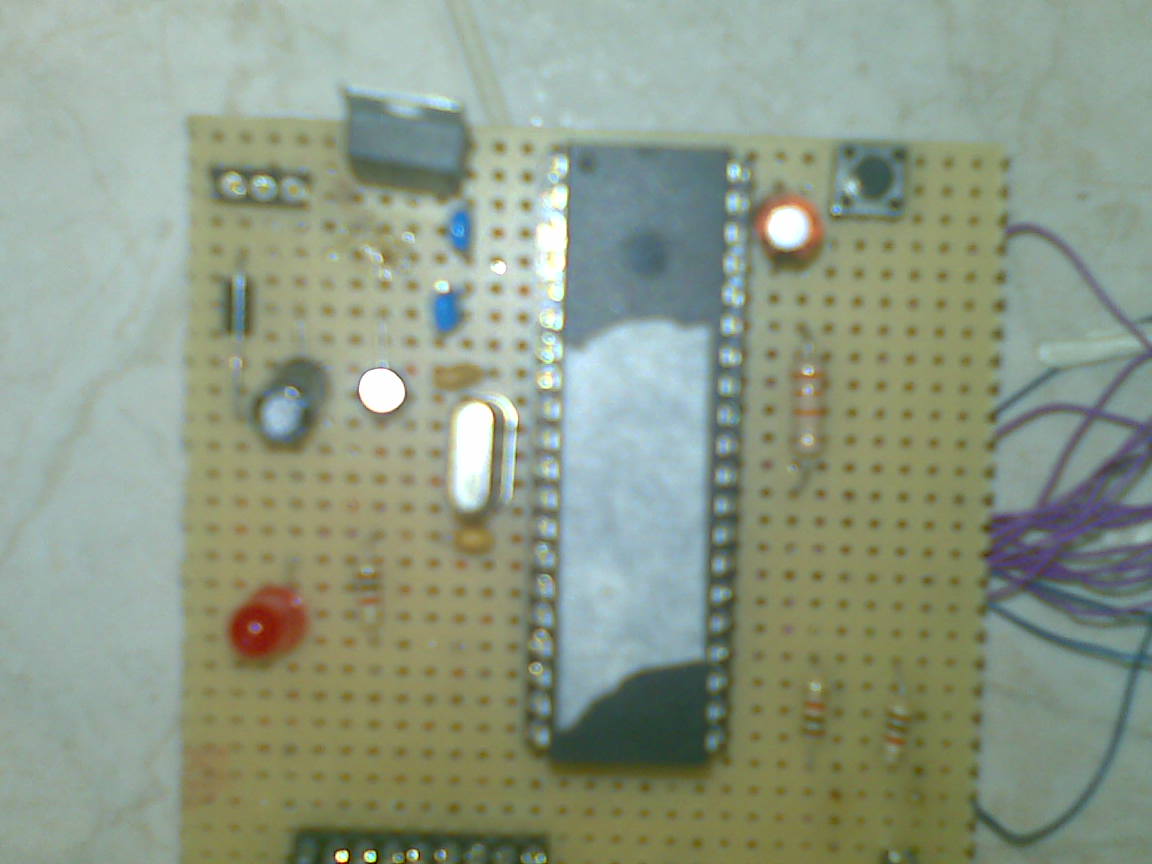
2\_ Capacitors:

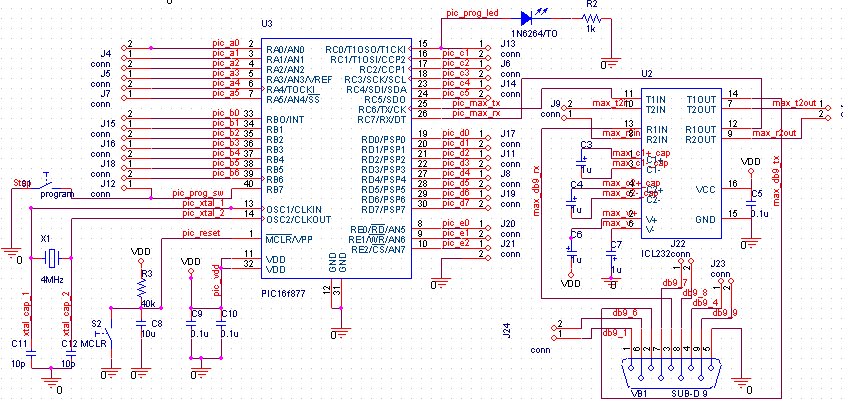
10Mf (1), 0.1Mf (1)

3\_Diode (1N4007)

**Connection Circuits:**

**Basic Circuit:**

****

****

**3\_Waiter's Controlling Motors Circuit (with the code):**

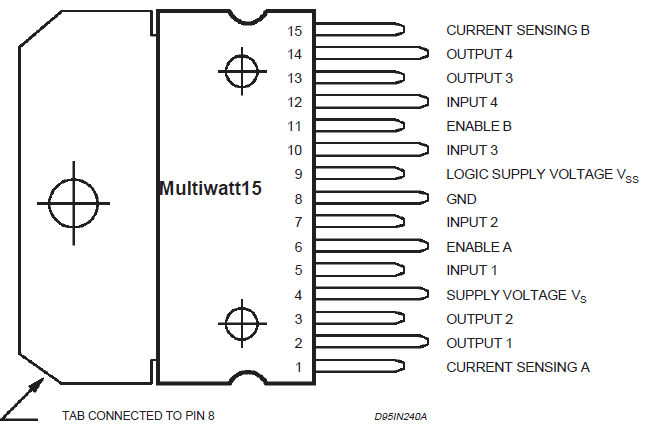
I used different circuits to control the motors direction and speed (but each one I faced a current problem until solve it by using relays, these circuits are :

**1\_HBridge (L298):**

**Description:**

The L298 is an integrated monolithic circuit in a 15-lead multi watt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals.

**Connection to the PIC:**

****

|  |  |
| --- | --- |
| PIC  Pins | HBridge  Pins |
| GND | 1 |
| Motor1 | 2 |
| Motor1 | 3 |
| 5V | 4 |
| CCP1 | 5 |
| A0 | 6 |
| CCP2 | 7 |
| GND | 8 |
| 5V | 9 |
| CCP1 | 10 |
| A1 | 11 |
| CCP2 | 12 |
| Motor2 | 13 |
| Motor2 | 14 |
| GND | 15 |

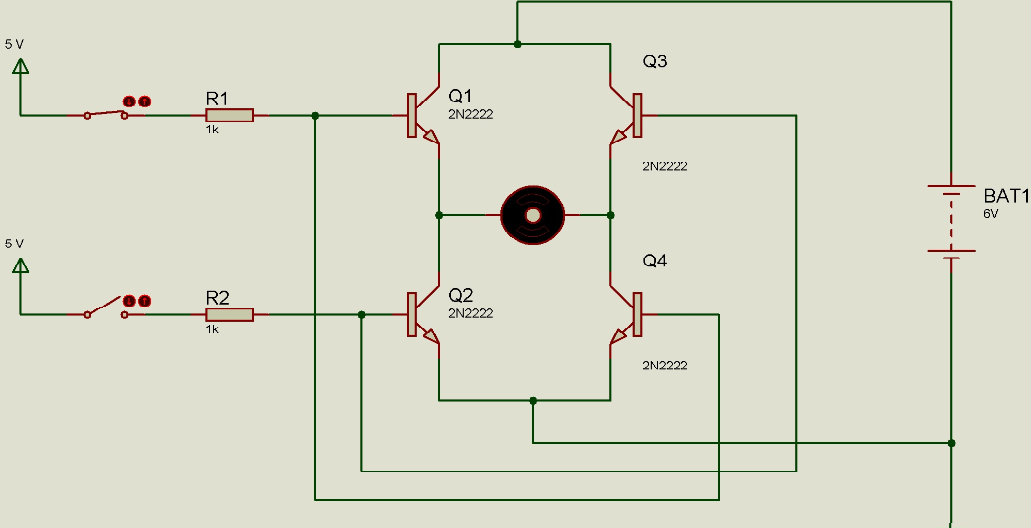
**The Problem of This Connection:**

1\_This connection work right for just one motor at a time,motor1 or motor2 for the same circuit at the same time if connect motor1 alone it worked if I connect motor2 alone also it worked but if I connect both motors it didn't work

2\_The current problem, this circuit made the motor needs 1.5 A or more to work (the waiter's motors), that mean need to replace the battery after run the motors 2 or 3 tests, which is not good.

So, I have to find another connection for my motors

**2\_Build HBridge by 4 Transistors :**

****

**Description:**

I tried to build HBridge by myself by using 4 transistors for control direction of motor. When 2 of transistors become ON (Q1 and Q4) the motor will rotate clockwise and when the 2 of transistors become ON (Q3 and Q2) the motor will rotate counter clockwise.

You make (Q1 and Q4) ON when close S.W1 by PIC (5 V), and make (Q3 and Q2) ON by PIC (5V).

**Connection to the PIC:**

**S.W1 ------------------🡪A0**

**S.W2------------------🡪A1**

When put (output\_high(pin\_a0)) that mean close S.W1 and make the motor turn clock wise, but when put (output\_high(pin\_a1)) that mean close S.W2 and make the motor turn counter clock wise.

**The Problem of This Connection:**

It worked well by change the direction of the motor , but the original problem still exist which is the current, the motor run slower and weaker than the time when I connect the motors directly to the batteries.

**3\_ULN (ULN2004):**

**Description:**

I tried here just to connect the 2 motors with PIC throw ULN, by just make on and off to the motors without control the direction of them, I tried to run one of them at a time and both of them at the same time.

**Connection to the PIC:**

|  |  |
| --- | --- |
| PIC  Pins | HBridge  Pins |
| A0 | 1 |
| A1 | 2 |
| GND | 8 |
| Motor1 | 15 |
| Motor2 | 16 |

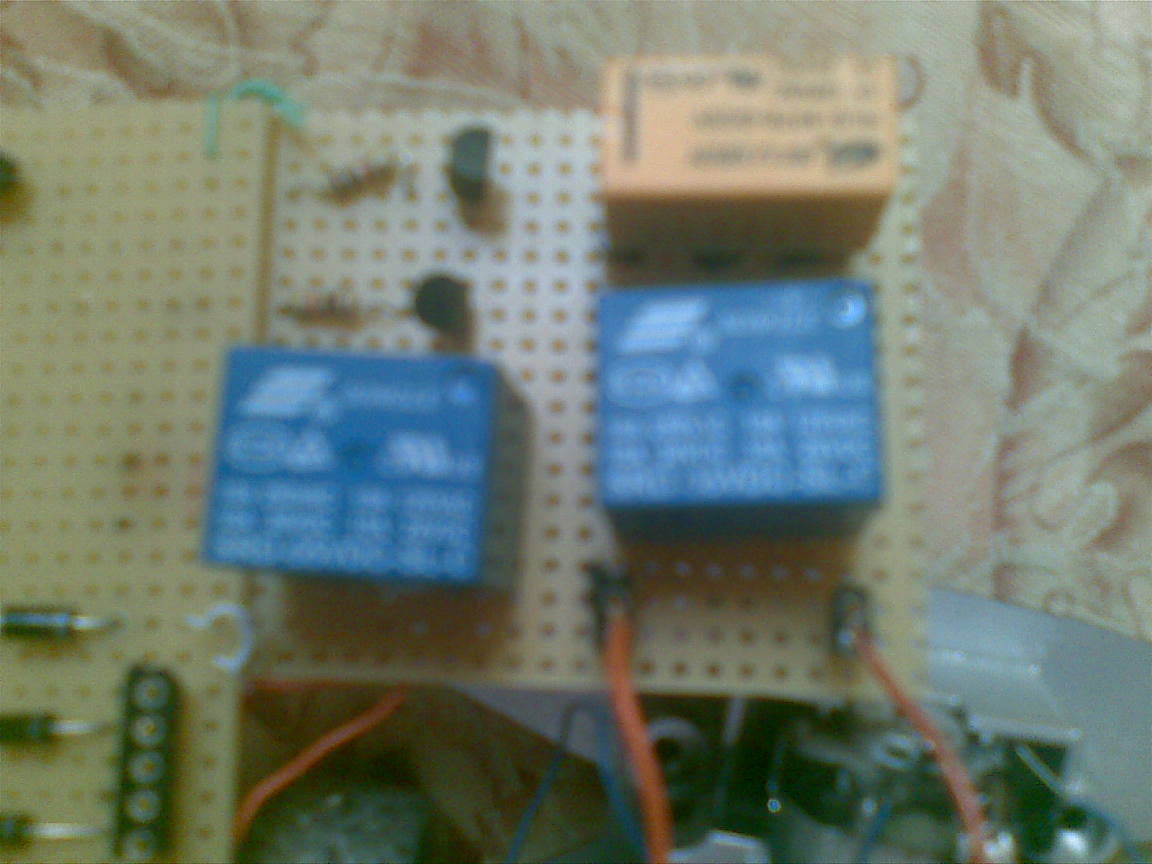
When put A0 (high) that mean connect one end of motor1 to the ground and the other end of the motor1 I connected before with (9V) that mean make motor1 run.

When put A1 (high) that mean connect one end of motor2 to the ground and the other end of the motor2 I connected before with (9V) that mean make motor2 run.

**The Problem of This Connection:**

**At Last I used relays which was the best solution for the current problem:**

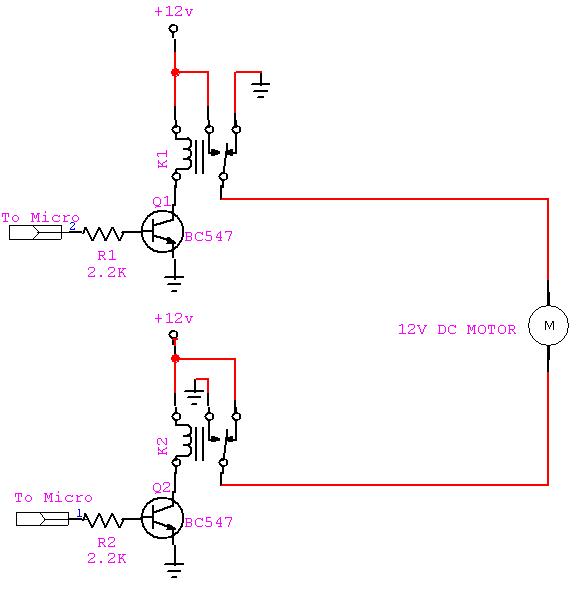
**Relays:**

****

**Description:**

The relays are mechanical switches that mean I connect the battery directly with the motor, that mean isolate the motors about the PIC. The motors don't take the current or the power from the PIC; I control the opening and closing the power to the motors.

**Connection to the PIC:**

****

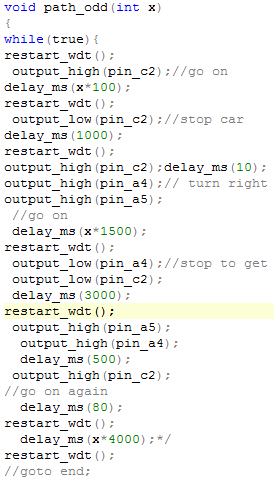
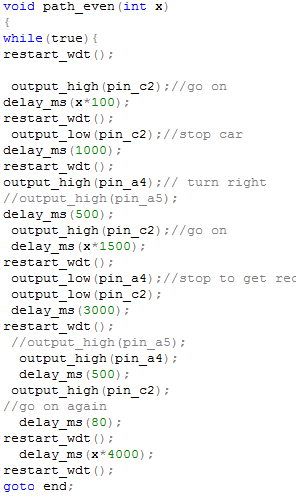
**Why This Circuit Worked Correctly?**

This circuit isolates the motors about the PIC. The motors don't take the current or the power from the PIC; I control the opening and closing the power to the motors.

**Finally:**

I connect 2 relays to control the front motor in direction and ON/OFF, and use 1 relay to control the back motor in ON/OFF.

**Code:**

****

**4\_Waiter's Body (LCD & Keypad) Circuit and code:**

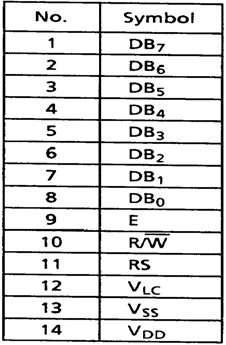
****

**The Components:**

**1\_LCD (M1632):** LIQUID CRYSTAL DISPLAY MODULE

****

**Datasheet:** I/O Terminal Symbols

****

**The Pins Connection to PIC:**

|  |  |  |
| --- | --- | --- |
| PIC Pins | LCD Pins name | LCD Pins |
| D7 | D7 | 1 |
| D6 | D6 | 2 |
| D5 | D5 | 3 |
| D4 | D4 | 4 |
| NC | NC | 5 |
| NC | NC | 6 |
| NC | NC | 7 |
| NC | NC | 8 |
| D0 | enable | 9 |
| D2 | rw | 10 |
| D1 | rs | 11 |
| GND | VLC | 12 |
| GND | VSS | 13 |
| 5V | VDD | 14 |

**The Code:**

If you connect the LCD in the right way, the code is very easy and work correctly. First but include to the LCD C file:

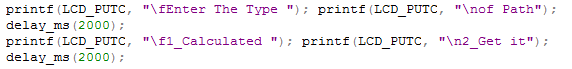
**#include <LCD.C>**

In the main function write this statement:

**lcd\_init ();**

Then to print anything, write this statement:

**Printf (LCD\_PUTC, "\hello");**

****

**2\_KEYPAD (3X4):**

****

**Datasheet:**

Here, you need to know which pins are for rows and which are for columns

Pins of keypad from left to right:

1--------🡪column2 2--------🡪row1

3--------🡪column1 4--------🡪row4

5--------🡪column3 6--------🡪row3

7--------🡪row2

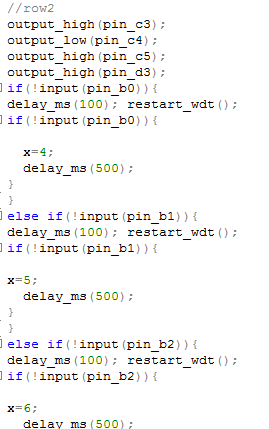
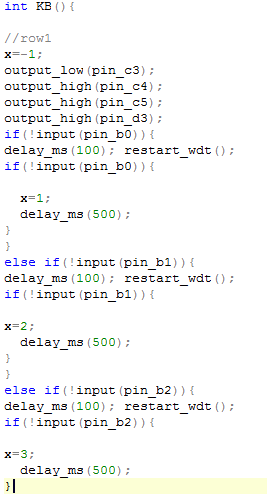
**Pins Connection to PIC:**

|  |  |
| --- | --- |
| Pins connection to PIC | Keypad Pins |
| B1 | 1 |
| C3 | 2 |
| B0 | 3 |
| D3 | 4 |
| B2 | 5 |
| C5 | 6 |
| C4 | 7 |

**The Code:**

The code here is the code of scanning to the keypad to listen to any pressed.

Don't forget to make debug for pressing the buttons



**The Goal of This Part:**

This part is designed to enter the data and know the response of the waiter in its different modes:

The waiter will know the mode that will run on by display the mode choices with its numbers on LCD. When the mode number entered by the keypad the chosen mode will run.

**1\_Add Mode:**

This mode is designed to add a new table to the restaurant. First the waiter wants to know the table's number.

After entering the number of table the waiter will check if that table is exist in its memory or not, if the table existed, a message will be display say that the table is existed and ask to a new table's number.

If the table is not existed, the waiter save the table number .Then the waiter wants to know if the path of going to that table is calculated or gets it by serial.

If the path is calculated: The waiter just saves flag (calculated).but if it is not calculated the waiter will ask to send the path of that table by serial which take the path from C# interface and save it on its memory. then the waiter show that the table is added .Then Back to ask the type of the mode want to run.

**2\_Get Order:**

In this mode, the waiter wait the number of table which received by wireless (IR) from the main controller, the waiter will check the table number if not exist in its memory will send error massage to main controller .If not, the waiter will check the type of path if calculated (The waiter will calculate it depends on the table number).If gets it by serial (The waiter will go to it depends on the path that saved on its memory).

After the waiter reach the table will ask the costumer to enter his order using LCD, after the costumer enter his order using keypad and enter finish, the waiter will back to its position.

**3\_Meet Order:**

Here, the waiter needs keypad just to enter the number of table to go to it to meet its order.

**5\_Waiter's Wireless (IR) (Transmitter and Receiver):**

At the waiter side, we have transmitter and receiver.

**The Transmitter (With Encoder):**

**Datasheet:**

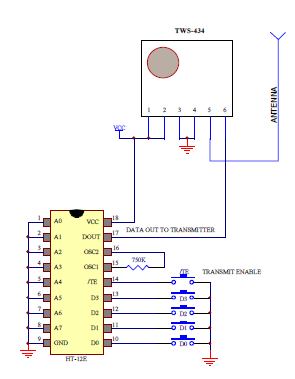
**Description**

The R F Solutions Ltd. AM hybrid transmitter module provides a complete RF transmitter which can be used to transmit data at up to 4KHz from any standard CMOS/TTL source.

The module is very simple to operate and offers low current consumption (typ. 4 mA). Data can be supplied directly from a microprocessor or encoding device, thus keeping the component count down and ensuring a low hardware cost.

****

**Pins Connection to PIC:**

****

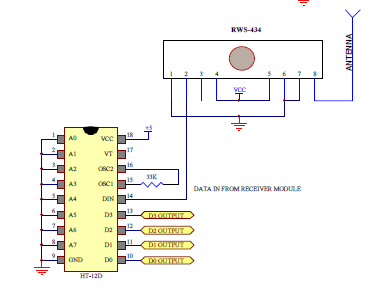
**The Receiver (With Decoder):**

**Datasheet:**

**Description**

****

**Pins Connection to PIC:**

****

**The Difficulties of Wireless Waiter:**

It made difficulties in code, but the major problem was the wireless damaged completely twice, so I stopped use them and started work on other wireless which is XBee.

**The New Wireless:**

**XBee:**

****

**Datasheet:**

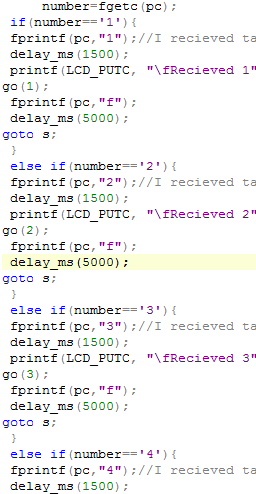
The [XBee/XBee-PRO ZB RF Modules](http://freedatasheets.com/datasheet-download/3987293560725436568bdd4ae72952a0/XBee%20%20XBee-PRO%20%20ZB%20RF%20Modules) are designed to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band. These are very powerful wireless modules that I have used extensively in projects; they are well worth talking a look at if you have not played with them in your latest wireless project.

**Connection to PIC:**

4 pins just need to connect

|  |  |
| --- | --- |
| Pins connection to PIC | XBee  Pins |
| VCC | VCC |
| GND | GND |
| A0(Trans) | DIN |
| A1(Rec) | Dout |

**Code:**

****

**The Main Controller:**

This part is easier than the previous part, no motors or mechanical base, but nothing can do easily. It has own problems that took a lot of time to solve them.

Let's look at the main controller's components in details:

**1\_The Basic Circuit:**

**Components:**

1\_PIC18F4620.

2\_Crystal (4MHZ)

3\_ Capacitors:

12pf (2), 6.8Mf (1), 100nf (3), 1Mf (4),

4\_Resistores:

33K (1), 1K (1)

5\_LED (1N6265)

6\_MAX (232)

7\_CONNECTOR DB9

8\_Bushbuttone (2)

**9\_Regulator Components:**

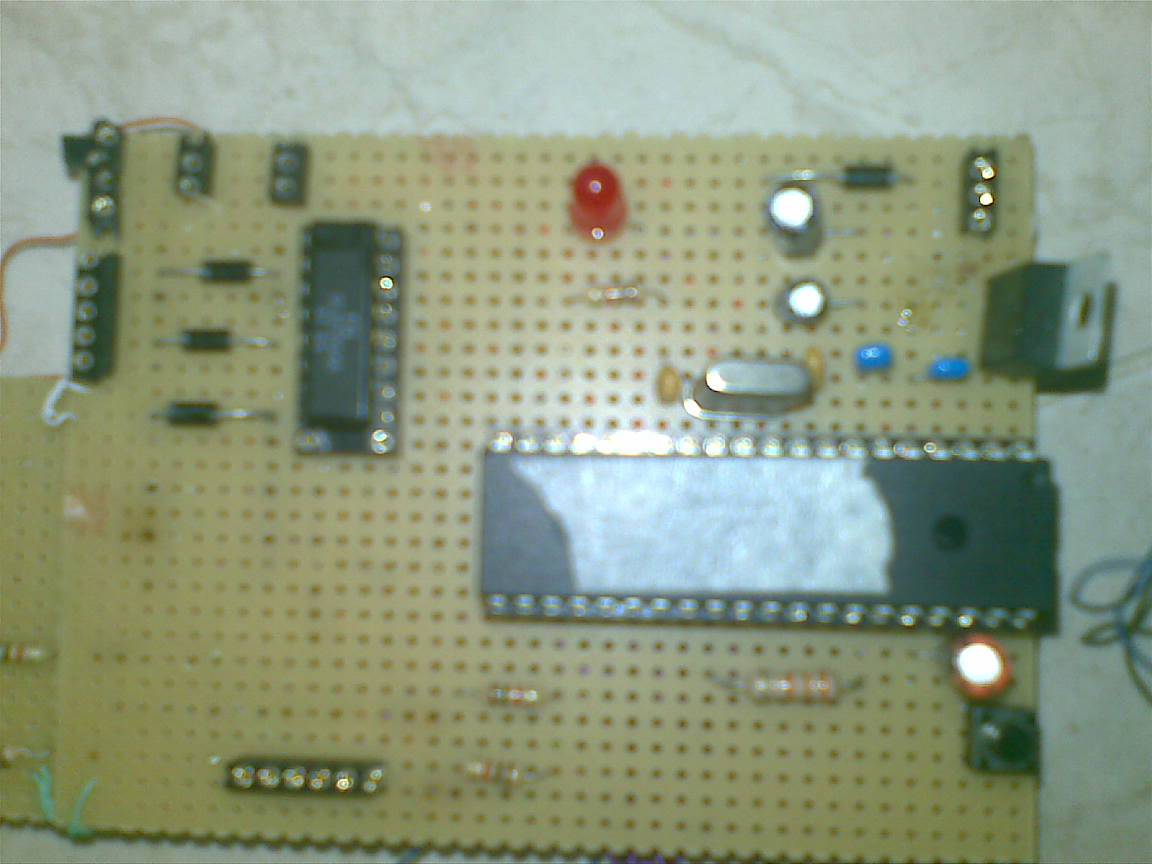
1\_LM7805C

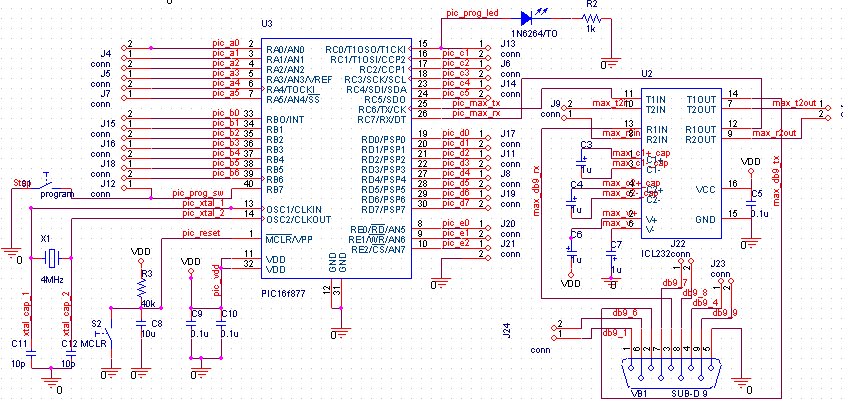
2\_ Capacitors:

10Mf (1), 0.1Mf (1)

3\_Diode (1N4007)

**Connection Circuit:**

****

** : 2\_The Wireless (IR) (Transmitter and Receiver)**

The same idea I discuss before I used XBee also here.

**The New Wireless:**

**XBee:**

****

**Datasheet:**

The [XBee/XBee-PRO ZB RF Modules](http://freedatasheets.com/datasheet-download/3987293560725436568bdd4ae72952a0/XBee%20%20XBee-PRO%20%20ZB%20RF%20Modules) are designed to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band. These are very powerful wireless modules that I have used extensively in projects; they are well worth talking a look at if you have not played with them in your latest wireless project.

**Connection to PIC:**

4 pins just need to connect

|  |  |
| --- | --- |
| Pins connection to PIC | XBee  Pins |
| VCC | VCC |
| GND | GND |
| A0(Trans) | DIN |
| A1(Rec) | Dout |

**Code:**

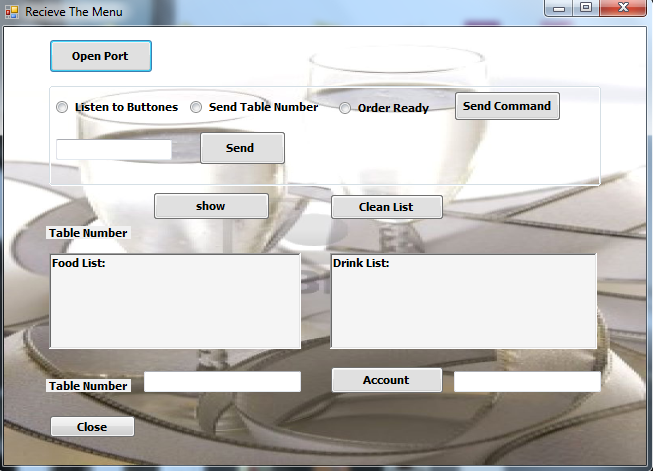
****

**C# Interface:**

****

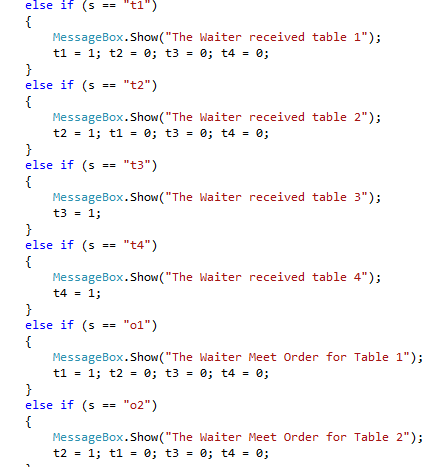
I did two parts for c# interface, these two parts are:

**1\_The Order Receiver and Cash Account:**

****

**Code:**

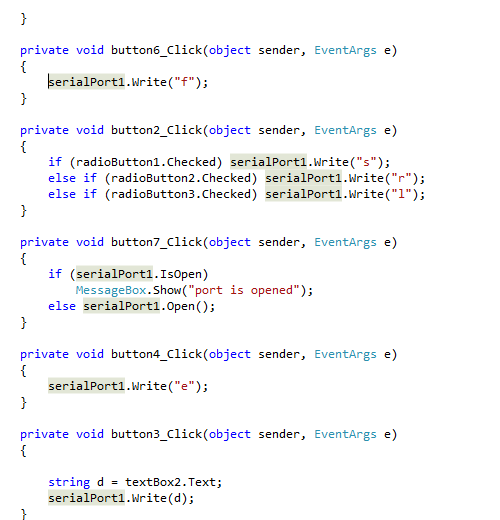
****

****

**2\_The Waiter's Path Sender:**

****

**Code:**

****

**The Problems I Faced in My Projects:**

1\_Mechanical Part.

2\_Motor’s Circuit.

3\_Wireless damaged in last 2 weeks.