Machine Learning in Robotics

N Jagadeeswar[1] and G V V Sharma[*]

Contents

***Abstract*—This manual provides an introduction of RF communication using arduino. It explains how to program using virtualwire library which is used for RF transmitter and receiver. This module tells us how to use RF communication in daily life.**

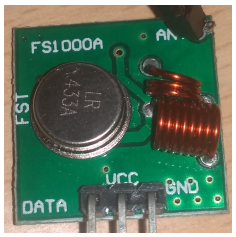

Fig. 1: Transmitter

Transmitter	data	Vcc	GND
Arduino	D12	5V	GND

- RF is one of the wireless communications.this RF communication limit upto 100mts.
- RF transmitter consists of three pins namely 1.data pin 2.Vcc 3.GND
- The data pin of transmitter is connected to the D12 of the Arduino, through this data pin only the data is transmitted.
- To activate the RF transmitter we connect Vcc to 5V of Arduino and GND to GND of the of the Arduino.

The author is intern of Summer Internship 2019 Program of IITH under Dr.G.V.V.Sharma with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India Email: [1], gadepall@iith.ac.in[*]

- Antenna is present at the corner.

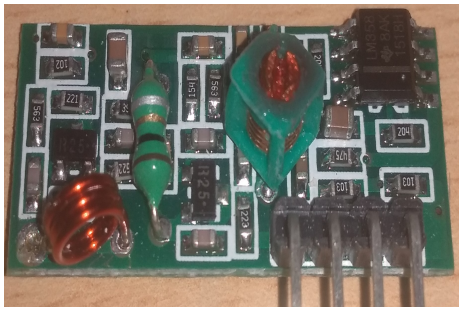

Fig. 2: Receiver

Receiver	data	Vcc	GND
Arduino	D11	5V	GND

- RF receiver part consists of same parts as in the Transmitter.
- At reciver we have two data pins we can use any one the two pins. Data pin is connected to the D11 of the Arduino.
- Remaining two connections are same as transmitter.

1 Controll of LEDs using RF

1.1 RF transmitter using pushbuttons

- For making keypad using push buttons connect the pushbuttons as in figure 3.
- Row pins R1,R2,R3 connected to the 2,3,4 pins of the Arduino respctively.
- Column pins start from left bottom corner and connect the consecutive column pins to the 5,6,7 of the arduino.

```
#include <VirtualWire.h>
#include <Keypad.h>
const byte ROWS=3;
const byte COLS=3;
char hexakeys[ROWS][COLS]={
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'}
```

```
};
byte rowpins[ROWS]={2,3,4};
byte colpins[COLS]={5,6,7};
Keypad customkeypad=Keypad(
   makeKeymap(hexakeys),rowpins,
   colpins,ROWS,COLS);

void setup()
{
  Serial.begin(9600);
    // Initialize the IO and ISR
  vw_setup(2000); // Bits per sec
}

void loop()
{
   char customKey=customkeypad.
      getKey();
  if(customKey)
  {
    Serial.println(customKey);
  }
  if (customKey=='1')
  {
    send("1");
    Serial.println("one");
  }
else if(customKey=='2')
  {
   send("2");
   Serial.println("two");
  }
  else if(customKey=='3')
  {
   send("3");
   Serial.println("three");
  }
  else if(customKey=='4')
{
  send("4");
  Serial.println("four");

}
else if(customKey=='5')
{
  send("5");
  Serial.println("five");

}
else if(customKey=='6')
{
  send("6");
  Serial.println("six");

}
else if(customKey=='7')
  {
   send("2");
   Serial.println("seven");
  }
else if(customKey=='8')
{
  send("8");
  Serial.println("eight");

}

  else if(customKey=='9')
  {
   send("2");
   Serial.println("nine");
  }
}

void send (char *message)
{
  vw_send((uint8_t *)message,
     strlen(message));
  vw_wait_tx(); // Wait until the
     whole message is gone
}
```

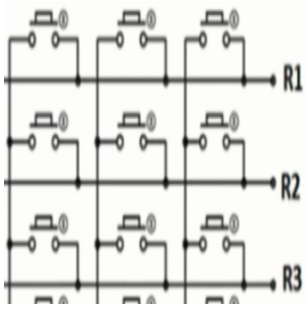

Fig. 3: keypad

1.2 RF receiver at LEDs

```
#include <VirtualWire.h>
const int led1 = 2;
const int led2 = 3;
const int led3 = 4;
```

```
const int led4 = 5;
const int led5 = 6;
const int led6 = 7;
const int led7 = 8;
const int led8 = 9;

byte message[VW_MAX_MESSAGE_LEN];
   // a buffer to store the
   incoming messages
byte messageLength =
   VW_MAX_MESSAGE_LEN; // the size
   of the message

void setup()
{
  Serial.begin(9600);
  Serial.println("Device is ready"
     );
 vw_setup(2000); // Bits per sec
  vw_rx_start(); // Start the
     receiver
pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
   pinMode(led6, OUTPUT);
    pinMode(led7, OUTPUT);
     pinMode(led8, OUTPUT);
}
void loop()
{
if (vw_get_message(message, &
   messageLength)) // Non-blocking
  {
    Serial.print("Received: ");
    for (int i = 0; i <
       messageLength; i++)
    {
      Serial.write(message[i]);


  if(message[i]=='2')
  {
    digitalWrite(led1,HIGH);
    Serial.println("led1");
  }
  else if(message[i]=='3')
  {
    digitalWrite(led2,HIGH);
  }
  else if(message[i]=='4')
  {
     digitalWrite(led3,HIGH);
  }
  else if (message[i]=='5')
  {
        digitalWrite(led4,HIGH);
  }
  else if(message[i]=='6')
  {
    digitalWrite(led5,HIGH);
  }
  else if(message[i]=='7')
  {
    digitalWrite(led6,HIGH);

  }
   else if(message[i]=='8')
  {
    digitalWrite(led7,HIGH);

  }
   else if(message[i]=='9')
  {
    digitalWrite(led8,HIGH);

    Serial.println();
}
 else if(message[i]=='1')
  {
    digitalWrite(led1,LOW);
digitalWrite(led2,LOW);
digitalWrite(led3,LOW);
digitalWrite(led4,LOW);
digitalWrite(led5,LOW);
   digitalWrite(led6,LOW);
digitalWrite(led7,LOW);
digitalWrite(led8,LOW);


}
Serial.println();
    }
  }
}
```

2 Controll of Toycar using RF

2.1 Joystick using RF

- Make the joystick using pushbuttons as in figure 2
- Red wires indicates rowpins and connect to 2,3,4 pins of the arduino respectively.
- Yellow wires indicates column pins connect from left corner to the 5,6,7 pins of the arduino.
- Connect the data pin of the transmitter to the D12 of the arduino. Connect Vcc to 5V and GND to GND of the arduino.
- Dump the following code. Your joystick is ready to use.

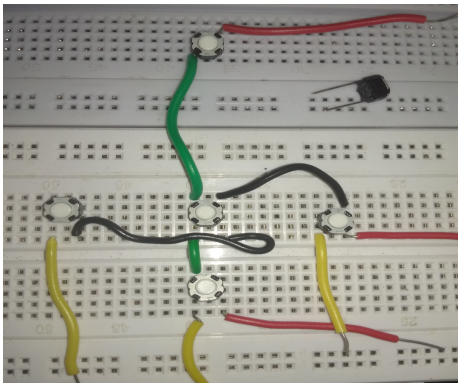

Fig. 4: keypad

```
#include <VirtualWire.h>
#include <Keypad.h>
#include<stdio.h>
const byte ROWS=3;
const byte COLS=3;
char hexakeys[ROWS][COLS]={
  { '␣','2','␣'},
  {'4','5','6'},
  {'␣','8','␣'}

};
byte rowpins[ROWS]={2,3,4};
byte colpins[COLS]={5,6,7};
Keypad customkeypad=Keypad(
   makeKeymap(hexakeys),rowpins,
   colpins,ROWS,COLS);

void setup()
{
  Serial.begin(9600);
    // Initialize the IO and ISR
  vw_setup(2000); // Bits per sec
}

void loop()
{
   char customKey=customkeypad.
      getKey();
  if(customKey)
  {
    Serial.println(customKey);
  }

  if(customKey=='2')
  {
   send("2");
   Serial.println("two");
  }
  else if(customKey=='4')
{
  send("4");
  Serial.println("four");

}
else if(customKey=='5')
{
  send("5");
  Serial.println("five");

}
else if(customKey=='6')
{
  send("6");
  Serial.println("six");

}
else if(customKey=='8')
{
  send("8");
  Serial.println("eight");

}

}

void send (char *message)
{
  vw_send((uint8_t *)message,
     strlen(message));
  vw_wait_tx(); // Wait until the
     whole message is gone
}
```

2.2 RF receiver at Toycar

- Make the toycar using the reference given below and connect the toy car.
- Always connect data pin of the receiver to D11 of the arduino.
- Use the following code and dump into the arduino.

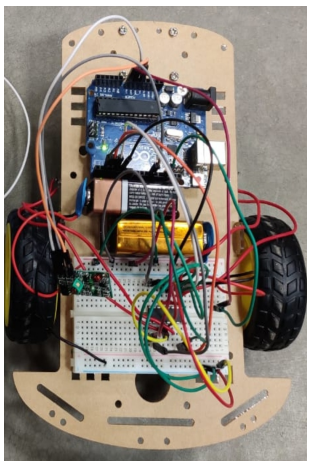

Fig. 5: toycar

```
#include <VirtualWire.h>
#include "Arduino.h"
const int m1n=3;
const int m2p=4;
const int m2n=5;
const int m1p=2;
byte message[VW_MAX_MESSAGE_LEN];
   // a buffer to store the
   incoming messages
byte messageLength =
   VW_MAX_MESSAGE_LEN; // the size
   of the message

void setup ( )
{
Serial.begin( 9600 ) ;
Serial.println("Device is ready");
  // Initialize the IO and ISR
  vw_setup(2000); // Bits per sec
  vw_rx_start(); // Start the
     receiver

pinMode ( m1p , OUTPUT) ;
pinMode ( m1n , OUTPUT) ;
pinMode ( m2p , OUTPUT) ;
pinMode ( m2n , OUTPUT) ;
}
void loop ( )
{
if (vw_get_message(message, &
   messageLength)) // Non-blocking
  {
    Serial.print("Received: ");
    for (int i = 0; i <
       messageLength; i++)
    {
      Serial.write(message[i]);


  if(message[i]==2)
  {
digitalWrite( m1p , HIGH) ;
digitalWrite( m1n , LOW) ;
digitalWrite( m2p , HIGH) ;
digitalWrite( m2n , LOW) ;
}
else if(message[i]==8)
{
//press 8
digitalWrite( m1p , LOW) ;
digitalWrite ( m1n , HIGH) ;
digitalWrite( m2p , LOW) ;
digitalWrite( m2n , HIGH) ;
}
else if(message[i]==4)
{
//press 4
digitalWrite( m1p , HIGH) ;
digitalWrite ( m1n , LOW) ;
digitalWrite( m2p , LOW) ;
digitalWrite( m2n , LOW) ;
}
else if (message[i]==6) {
//press 6
digitalWrite( m1p , LOW) ;
digitalWrite ( m1n , LOW) ;
digitalWrite( m2p , HIGH) ;
digitalWrite( m2n , LOW) ;
}
else if(message[i]==5)
{
//press 5
digitalWrite( m1p , LOW) ;
digitalWrite( m1n , LOW) ;
```

```
digitalWrite( m2p , LOW) ;
digitalWrite( m2n , LOW) ;
}
  }
    Serial.println();
}
}
```

References

[1] https://github.com/gadepall/EE1320/blob/master/toycar/gvv$_t$*oycar.pdf*