//WRITTEN BY: VIDUR NAYYAR

//DESIGNED BY: VIDUR NAYYAR

//DEBUGGED BY: VIDUR NAYYAR

//TESTED BY: VIDUR NAYYAR

//DATE : 12/14/13

// ALL RIGHTS TO THE CODE RESERVED

#include <VirtualWire.h>

//char msgC[1]={'C','',''};

char fb1, fb2;

int alarm=43;

int returned;

// Buffer to store incoming commands from serial port

String input\_command;

void setup()

{

pinMode(alarm,OUTPUT);

Serial.begin(9600);// testing and debugging

Serial.println("setup....");// debugging

//Initialise the IO and ISR

vw\_set\_tx\_pin(26);// pin 3 is used for transmitting data out of the TX Link module

vw\_set\_rx\_pin(7);// pin 4 is used for receiving data in from the RX Link module

//vw\_set\_ptt\_inverted(true);//Required for DR3100

vw\_setup(2000);// bits per second

vw\_rx\_start();// start the receiver PLL running

}

void loop()

{

// by this we can receive any command over serial input and it will be accepted as long as it has a "#" at its end

while (Serial.available()>0) // if any command is available on the serial port

{

int encode;

char received = Serial.read();// read the command and store the bits

input\_command += received; //pusing the command up the buffer

if (received =='#')

{

Serial.println("I received:");// debug and test

Serial.print(input\_command);// debug and test

char a= input\_command[1];

if(input\_command[0]=='A') // for command unit 1

{

switch (a)

{

case 'A':

Serial.println("Turn light ON at 100%");

encode = 0;

break;

case 'B':

Serial.println("Turn light OFF at 0%");

encode = 1;

break;

case 'C':

Serial.println("Turn light ON at 25%");

encode = 2;

break;

case 'D':

Serial.println("Turn light ON at 50%");

encode = 3;

break;

case 'E':

Serial.println("Turn light ON at 75%");

encode = 4;

break;

case 'F':

Serial.println("Feedback");

encode = 5;

break;

}}

if(input\_command[0]=='B')

{

switch (a)

{

case 'A':

Serial.println("Increase volume to 100%");

encode = 6;

break;

case 'B':

Serial.println("Decrease volume to 0%");

encode = 7;

break;

case 'C':

Serial.println("Change volume to 25%");

encode = 8;

break;

case 'D':

Serial.println("Change volume to 50%");

encode = 9;

break;

case 'E':

Serial.println("Change volume to 75%");

encode = 10;

break;

case 'F':

Serial.println("Turn OFF the music player");

encode = 11;

break;

case 'G':

Serial.println("Turn ON the music player");

encode = 12;

break;

}

}

input\_command = "";// Clear recieved buffer

tx(encode);

} //not sure if we should 1st fill the buffer then go ahead or how

}

}

// transmitting the encoded command to the controllers....

// can also say if(a<6)--->

void tx(int a)

{

char msgA[2]={'A',' '};

char msgB[2]={'B',' '};

Serial.println("comand is:");

Serial.println(a);

if (a==0)

{

msgA[1] = 'A';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==1)

{

//msgA[0] = 'A';

msgA[1] = 'B';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==2)

{

// msgA[0] = 'A';

msgA[1] = 'C';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==3)

{

msgA[1] = 'D';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==4)

{

//msgA[0] = 'A';

msgA[1] = 'E';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==5)

{

//msgA[0] = 'A';

msgA[1] = 'F';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgA);

vw\_send((uint8\_t\*)msgA, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The Light is glowing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==6)

{

msgB[1] = 'A';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==7)

{

msgB[1] = 'B';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==8)

{

msgB[1] = 'C';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==9)

{

msgB[1] = 'E';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==10)

{

msgB[1] = 'E';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==11)

{

msgB[1] = 'G';// we skipped F for feedback

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

if(returned==0)

{

Serial.println("The music player is OFF");

}

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==12)

{

msgB[1] = 'H';

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

if(returned>25)

{

Serial.println("Music Player is ON");

}

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

/\*

if (a==13)

{

int y=0;//reset y

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(1000);//wait time timeout\_ms

char fb1= rx1();// feedback1

if((x==1) &(fb1!='Z'))

{

Serial.print("fb1");//debug

Serial.println(fb1);

returned= brightness\_case(fb1);

Serial.print(returned);

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0;

Serial.println("alarm off");

//k=50;

do

{

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

j= rx();// end sign if j!=0 it will go out of while loop

//k--;// keep checking till 100 iterations then snap out

// if (k==0)

//{

//j=1;

//}

}

while(j=0);

Serial.println("The music is playing at:");

// Serial.print("fb1 val");

//Serial.println(fb1);// we can also make a switch to map fb to value

//returned= brightness\_case(fb1);

Serial.print("returned:");

if(returned==0)

{

Serial.print("Music player is OFF");

}

Serial.print(returned);//debug

Serial.print("%");

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}

if (a==14)

{

msgB[1] = 'F';

Serial.println(msgB);

int x=0; // reset the value of x

//receiver feedback and handshake mechanism starts here

for(int i=10;i>0; i--) // repeat the command send 10 times if no reception is acheived

// then snap out of it and raise an alarm

{

Serial.println(msgB);

vw\_send((uint8\_t\*)msgB, 2);//$$$$$%%%%%%%$@@@@@ dont know as yet 2 or 3 or 1

vw\_wait\_tx();

x=vw\_wait\_rx\_max(100);//wait time timeout\_ms

if(x==1)

{

i=0; // snap out of the loop if reception is there and a true signal is acheived

}

}

// now if there was a reception

if(x==1)

{

digitalWrite(alarm,LOW); //reset alarm (if some alarm was high in other command)

int j=0,k=0;

k=1000;

do

{

char fb1= rx1();// feedback1

delay((20));

j= rx();// end sign if j!=0 it will go out of while loop

k--;// keep checking till 100 iterations then snap out

if (k==0)

{

j=1;

}

}

while(j=0);

Serial.print("The music is playing at:");

Serial.print(fb1);// we can also make a switch to map fb to value

int returned= brightness\_case(fb1);

Serial.print(returned);

k =1000;

do

{

char fb2= rx1();// feedback 2

int j= rx();//end sign

k--;

if (k==0)

{

j=1;

}

}

while(j=0);

Serial.print("updated feedback: Now the music is playing at:");

Serial.print(fb2);// we can also make a switch to map fb to value

returned= brightness\_case(fb2);

Serial.print(returned);

}

else

digitalWrite(alarm,HIGH); //raise alarm if x!=1

}\*/

}

char rx1()// for the feedback

{

char return\_fb;

uint8\_t buf[VW\_MAX\_MESSAGE\_LEN];

uint8\_t buflen = VW\_MAX\_MESSAGE\_LEN;

if (vw\_get\_message(buf, &buflen)) // Non-blocking

{

int i;

// Message with a good checksum received, dump it.

Serial.print("Got: ");

// only for debug

for (i = 0; i < 2; i++)

{

Serial.print((char)buf[i]);

Serial.print(" ");

}

if(buf[0]=='A')

{

Serial.print(return\_fb);

if(buf[1]!='Z')

{

return\_fb = buf[1];

}

}

else return\_fb = 'Z';

}

Serial.print("return\_fb");

Serial.print(return\_fb);//debug

return return\_fb;

}

int rx()// for just checking the end of message

{

uint8\_t buf[VW\_MAX\_MESSAGE\_LEN];

uint8\_t buflen = VW\_MAX\_MESSAGE\_LEN;

int return\_value;

if (vw\_get\_message(buf, &buflen)) // Non-blocking

{

int i;

// Message with a good checksum received, dump it.

Serial.print("Got: ");

// only for debug

for (i = 0; i < buflen; i++)

{

Serial.print(buf[i]);

Serial.print(" ");

}

if((buf[0]=='Z') && (buf[1]=='Z') )// in case message is of ending

{

return\_value=1;

}

else

{

return\_value=0;

}

}

return return\_value;

}

int brightness\_case(char fb)

{

int brightness;

switch(fb) // now we are only interested in second buff position

{

case 'A'://for 100% brightness

Serial.println("100% brightness/ volume");

brightness = 100;

break;

case 'B':// for 0% brightness

Serial.println("0% brightness/ volume");

brightness =0;

break;

case 'C':// for 25% brightness

Serial.println("25% brightness/ volume");

brightness = 25;

break;

case 'D':

Serial.println("50% brightness/ volume");

brightness = 50;

break;

Serial.print("brightness from case=");

Serial.println(brightness);// debug

return brightness;

}

}