/\* 9-10-11

\*A008

Pat's BT-IR Hexapod-12- Finger On to Run, Off to Stop.

This Combo BT-IR Sketch is controlled by an Android Phone, IR Remote M and Mini Remote.

There was a conflict using PWM pin 11 when you combined both BT & IR, Library analogWrite with pin 11?

Modified by Pat McMahon (V8) 5/8/2021, servo1=9,servo2=10,servo3=11, added Front pin 3, Middle extra pin 13,Back Lights pin 4,HORN pin 12.

Adjusted the code plus for combinations of backleft, backright,frontleft,frontright.

\*/

#include <IRremote.h>

#include <Servo.h>

const int frontLeds\_pin = 3;

const int backLeds\_pin = 4;

const int centreLeds = 13;

const int buzzer = 12;

Servo servo1; // create servo object to control a servo

Servo servo2; // create servo object to control a servo

Servo servo3; // create servo object to control a servo

const int RECV\_PIN = 2;

const int SEND\_PIN = 12;

#define USE\_NO\_CARRIER = 1;

int lastMiniIRCommand = 0;

int pos = 0; // variable to store the servo position

IRrecv irrecv(RECV\_PIN, SEND\_PIN);

const int delayTime = 300;

#define FRONTLIGHTS\_BLUELED 3 //Blue LED's

#define BACKLIGHTS\_REDLED 4 //Red Led's

#define HORN 12 //Passive Buzzer

#define EXTRA\_ORANGELED 13 //Yellow Led's

int command; //Int to store app command state.

int Speed = 204; // 0 - 255.

int Speedsec;

int buttonState = 0;

int lastButtonState = 0;

int Turnradius = 0; //Set the radius of a turn, 0 - 255 Note:the robot will malfunction if this is higher than int Speed.

int brakeTime = 45;

int brkonoff = 1; //1 for the electronic braking system, 0 for normal.

void setup()

{

servo1.attach(9); // attaches the servo on pin 9 to the servo object

servo2.attach(10); // attaches the servo on pin 9 to the servo object

servo3.attach(11); // attaches the servo on pin 9 to the servo object

irrecv.enableIRIn(); // Start the receiver

Serial.begin(9600); //Set the baud rate to your Bluetooth module.

Servo servo1; // create servo object to control a servo

Servo servo2; // create servo object to control a servo

Servo servo3; // create servo object to control a servo

// pinMode(LED, OUTPUT); //Set the LED pin.

pinMode(FRONTLIGHTS\_BLUELED, OUTPUT);

pinMode(BACKLIGHTS\_REDLED, OUTPUT);

pinMode(HORN, OUTPUT);

pinMode(EXTRA\_ORANGELED, OUTPUT);// MIDDLELIGHTS

pinMode(LED\_BUILTIN, OUTPUT);

Serial.begin(9600); //Set the baud rate to your Bluetooth module.

pinMode(frontLeds\_pin, OUTPUT);

pinMode(backLeds\_pin, OUTPUT);

pinMode(centreLeds, OUTPUT);

pinMode(buzzer, OUTPUT);

irrecv.enableIRIn(); // Start the receiver

}

void loop() {

if (Serial.available() > 0)

{

command = Serial.read();

// Stop(); //Initialize with motors stoped.

switch (command)

{

case 'F':

forwards();

break;

case 'B':

backwards();

break;

case 'L':

left();

break;

case 'R':

right();

break;

case 'G':

forwardleft();

break;

case 'I':

forwardright();

break;

case 'H':

backleft();

break;

case 'J':

backright();

break;

case 'W':

FrontLightsOn();

break;

case 'w':

FrontLightsOff();

break;

case 'U':

BackLightsOn();

break;

case 'u':

BackLightsOff();

break;

case 'V':

HornOn();

break;

case 'v':

HornOff();

break;

case 'X':

ExtraOn();

break;

case 'x':

ExtraOff();

break;

Speed = 100;

break;

case '1':

Speed = 140;

break;

case '2':

Speed = 153;

break;

case '3':

Speed = 165;

break;

case '4':

Speed = 178;

break;

case '5':

Speed = 191;

break;

case '6':

Speed = 204;

break;

case '7':

Speed = 216;

break;

case '8':

Speed = 229;

break;

case '9':

Speed = 242;

break;

case 'q':

Speed = 255;

break;

}

}

decode\_results results;

if (irrecv.decode(&results)) // if there is an IR reading

{

Serial.println(results.value, HEX);

{

switch (results.value)

{

//Remote M

case 0x2F0:

Serial.println("Forwards!");

forwards();

break;

case 0xAF0:

Serial.println("Backwards!");

backwards();

break;

case 0x2D0:

Serial.println("Lefty!");

left();

break;

case 0xCD0:

Serial.println("Righty!");

right();

break;

case 0x738:

Serial.println("centreLeds!"); // Middle Orange Led's

middleLeds();

break;

case 0xF38:

Serial.println("frontLeds!"); // Front Blue Led's

frontLeds();

break;

case 0x338:

Serial.println("backLeds!"); //Back Red Led's

backLeds();

break;

case 0xB38:

Serial.println("horn!");

HornOn;

break;

case 0xA70:

Serial.println("Stop!");

;

break;

}

}

/\*//Mini NEC Remote

{

case 0xFF18E7:

Serial.println("Forwards!");

forwards();

lastMiniIRCommand = results.value;

break;

case 0xFF4AB5:

Serial.println("Backwards!");

backwards();

lastMiniIRCommand = results.value;

break;

case 0xFF10EF:

Serial.println("Lefty!");

lefty();

lastMiniIRCommand = results.value;

break;

case 0xFF5AA5:

Serial.println("Righty!");

righty();

lastMiniIRCommand = results.value;

break;

case 0xFF38C7:

Serial.println("Stop!");

halt();

lastMiniIRCommand = results.value;

break;

case 0xFF629D:

Serial.println("centreLeds!"); // Middle Orange Led's

centreLeds();

lastMiniIRCommand = results.value;

break;

case 0xFFA25D:

Serial.println("frontLeds!"); // Front Blue Led's

frontLeds();

lastMiniIRCommand = results.value;

break;

case 0xFFE21D:

Serial.println("backLeds!"); //Back Red Led's

backLeds();

lastMiniIRCommand = results.value;

break;

case 0xFF02FD:

Serial.println("horn!");

horn();

lastMiniIRCommand = results.value;

break;

case 0xFFFFFFFF:

Serial.println("RepeatLastCommand!");

Serial.println(lastMiniIRCommand, HEX);

switch (lastMiniIRCommand)

case 0xFF18E7:

Serial.println("Repeat Forwards!");

forwards();

break;

case 0xFF4AB5:

Serial.println("Repeat Backwards!");

backwards();

break;

case 0xFF10EF:

Serial.println("Repeat Left!");

left();

break;

case 0xFF5AA5:

Serial.println("Repeat Right!");

right();

break;

case 0xFF38C7:

Serial.println("Repeat Stop!");

halt();

break;

}

//\*/

irrecv.resume(); // Receive the next value

}

}

//BT Commands

void forwards()

{

Serial.println("BT Forward");

digitalWrite(FRONTLIGHTS\_BLUELED, HIGH);

servo1.write(105);

delay(100);

servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(75);

delay(100);

servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void backwards()

{

Serial.println("BT Back");

digitalWrite(BACKLIGHTS\_REDLED, HIGH);

servo1.write(75);

delay(100);

servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(105);

delay(100);

servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void left()

{

Serial.println("BT Left");

servo1.write(105);

delay(100);

//servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(75);

delay(100);

//servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

delay(100);

}

void right()

{

Serial.println("BT Right");

servo1.write(75);

delay(100);

servo2.write(75);

//servo3.write(105);

delay(100);

servo1.write(105);

delay(100);

servo2.write(105);

// servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void forwardleft()

{

}

void forwardright()

{

}

void backright()

{

}

void backleft()

{

}

void FrontLightsOff()

{

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

}

void FrontLightsOn()

{

digitalWrite(FRONTLIGHTS\_BLUELED, HIGH);

}

void HornOff()

{

digitalWrite(HORN, LOW);

}

void HornOn()

{

digitalWrite(HORN, HIGH);

}

void ExtraOff()

{

digitalWrite(EXTRA\_ORANGELED, LOW);

}

void ExtraOn()

{

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

delay(500); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

delay(500); // wait for a second

}

void BackLightsOff()

{

digitalWrite(BACKLIGHTS\_REDLED, LOW);

}

void BackLightsOn()

{

digitalWrite(BACKLIGHTS\_REDLED, HIGH);

}

void Stop()

{

}

void middleLeds()

{

digitalWrite(EXTRA\_ORANGELED, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(EXTRA\_ORANGELED, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

// lastButtonState = buttonState();

// (

// }

void brakeOff()

{

}

//IR Commands

void irforwards()

{

Serial.println("IRforwards");

//forwards();

digitalWrite(FRONTLIGHTS\_BLUELED, HIGH);

servo1.write(105);

delay(100);

servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(75);

delay(100);

servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void irbackwards()

{

Serial.println("IRbackwards");

//backwards();

digitalWrite(BACKLIGHTS\_REDLED, HIGH);

servo1.write(75);

delay(100);

servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(105);

delay(100);

servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void righty()

{

Serial.println("IRrighty");

//righty();

servo1.write(75);

delay(100);

servo2.write(75);

//servo3.write(105);

delay(100);

servo1.write(105);

delay(100);

servo2.write(105);

// servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void lefty()

{

Serial.println("IRlefty");

//lefty();

servo1.write(105);

delay(100);

//servo2.write(105);

servo3.write(105);

delay(100);

servo1.write(75);

delay(100);

//servo2.write(75);

servo3.write(75);

delay(100);

servo1.write(90);

servo2.write(90);

servo3.write(90);

delay(100);

}

void irmiddleLeds()

{

// turn on EXTRA\_ORANGELED

digitalWrite(EXTRA\_ORANGELED, HIGH);

delay(delayTime);

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

digitalWrite(BACKLIGHTS\_REDLED, LOW);

digitalWrite(EXTRA\_ORANGELED, LOW);

digitalWrite(buzzer, LOW);

}

void frontLeds(void)

{

// turn on FRONTLIGHTS\_BLUELED

digitalWrite(FRONTLIGHTS\_BLUELED, HIGH);

delay(delayTime);

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

digitalWrite(BACKLIGHTS\_REDLED, LOW);

digitalWrite(EXTRA\_ORANGELED, LOW);

digitalWrite(buzzer, LOW);

}

void backLeds()

{

// turn on BACKLIGHTS\_REDLED

digitalWrite(BACKLIGHTS\_REDLED, HIGH);

delay(delayTime);

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

digitalWrite(BACKLIGHTS\_REDLED, LOW);

digitalWrite(EXTRA\_ORANGELED, LOW);

digitalWrite(buzzer, LOW);

}

void horn()

{

// turn on horn

digitalWrite(buzzer, HIGH);

delay(delayTime);

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

digitalWrite(BACKLIGHTS\_REDLED, LOW);

digitalWrite(EXTRA\_ORANGELED, LOW);

digitalWrite(buzzer, LOW);

}

void halt()

{

digitalWrite(FRONTLIGHTS\_BLUELED, LOW);

digitalWrite(BACKLIGHTS\_REDLED, LOW);

digitalWrite(EXTRA\_ORANGELED, LOW);

digitalWrite(buzzer, LOW);;

}