www.osoyoo.com IR remote control smart car

```
 * program tutorial http://osoyoo.com/2018/12/14/osoyoo-smart-car-lesson-2-control-robot-car-through-infrared-remote/
 */
#include <IRremote.h>
#define IR_PIN    10 //IR receiver Signal pin connect to Arduino pin D2
 IRrecv IR(IR_PIN);  //   IRrecv object  IR get code from IR remoter
 decode_results IRresults;
#define speedPinR 9    //  RIGHT PWM pin connect MODEL-X ENA
#define RightDirectPin1  12    //Right Motor direction pin 1 to MODEL-X IN1
#define RightDirectPin2  11    //Right Motor direction pin 2 to MODEL-X IN2
#define speedPinL 6    // Left PWM pin connect MODEL-X ENB
#define LeftDirectPin1  7    //Left Motor direction pin 1 to MODEL-X IN3
#define LeftDirectPin2  8   //Left Motor direction pin 1 to MODEL-X IN4

 #define IR_ADVANCE      0x00FF18E7      //code from IR controller "▲" button
 #define IR_BACK         0x00FF4AB5      //code from IR controller "▼" button
 #define IR_RIGHT        0x00FF5AA5      //code from IR controller ">" button
 #define IR_LEFT         0x00FF10EF      //code from IR controller "<" button
 #define IR_STOP         0x00FF38C7      //code from IR controller "OK" button
 #define IR_turnsmallleft 0x00FFB04F      //code from IR controller "#" button

enum DN
{
  GO_ADVANCE, //go forward
  GO_LEFT, //left turn
  GO_RIGHT,//right turn
  GO_BACK,//backward
```

```
  STOP_STOP,
  DEF
}Drive_Num=DEF;

bool stopFlag = true;//set stop flag
bool JogFlag = false;
uint16_t JogTimeCnt = 0;
uint32_t JogTime=0;
uint8_t motor_update_flag = 0;
/***************motor control***************/
void go_Advance(void)  //Forward
{
  digitalWrite(RightDirectPin1, HIGH);
  digitalWrite(RightDirectPin2,LOW);
  digitalWrite(LeftDirectPin1,HIGH);
  digitalWrite(LeftDirectPin2,LOW);
  analogWrite(speedPinL,255);
  analogWrite(speedPinR,255);
}
void go_Left(int t=0)  //Turn left
{
  digitalWrite(RightDirectPin1, HIGH);
  digitalWrite(RightDirectPin2,LOW);
  digitalWrite(LeftDirectPin1,LOW);
  digitalWrite(LeftDirectPin2,HIGH);
  analogWrite(speedPinL,200);
  analogWrite(speedPinR,200);
  delay(t);
}
```

```
void go_Right(int t=0)  //Turn right
{
  digitalWrite(RightDirectPin1, LOW);
  digitalWrite(RightDirectPin2,HIGH);
  digitalWrite(LeftDirectPin1,HIGH);
  digitalWrite(LeftDirectPin2,LOW);
  analogWrite(speedPinL,200);
  analogWrite(speedPinR,200);
  delay(t);
}
void go_Back(int t=0)  //Reverse
{
  digitalWrite(RightDirectPin1, LOW);
  digitalWrite(RightDirectPin2,HIGH);
  digitalWrite(LeftDirectPin1,LOW);
  digitalWrite(LeftDirectPin2,HIGH);
  analogWrite(speedPinL,255);
  analogWrite(speedPinR,255);
  delay(t);
}
void stop_Stop()    //Stop
{
  digitalWrite(RightDirectPin1, LOW);
  digitalWrite(RightDirectPin2,LOW);
  digitalWrite(LeftDirectPin1,LOW);
  digitalWrite(LeftDirectPin2,LOW);
}

/**************detect IR code***************/
```

```
void do_IR_Tick()
{
  if(IR.decode(&IRresults))
  {
    if(IRresults.value==IR_ADVANCE)
    {
      Drive_Num=GO_ADVANCE;
    }
    else if(IRresults.value==IR_RIGHT)
    {
       Drive_Num=GO_RIGHT;
    }
    else if(IRresults.value==IR_LEFT)
    {
       Drive_Num=GO_LEFT;
    }
    else if(IRresults.value==IR_BACK)
    {
       Drive_Num=GO_BACK;
    }
    else if(IRresults.value==IR_STOP)
    {
       Drive_Num=STOP_STOP;
    }
    IRresults.value = 0;
    IR.resume();
  }
}
```

```
/**************car control**************/
void do_Drive_Tick()
{
  switch (Drive_Num)
  {
    case GO_ADVANCE:go_Advance();JogFlag = true;JogTimeCnt = 1;JogTime=millis();break;//if GO_ADVANCE code is detected, then go advance
    case GO_LEFT: go_Left();JogFlag = true;JogTimeCnt = 1;JogTime=millis();break;//if GO_LEFT code is detected, then turn left
    case GO_RIGHT:  go_Right();JogFlag = true;JogTimeCnt = 1;JogTime=millis();break;//if GO_RIGHT code is detected, then turn right
    case GO_BACK: go_Back();JogFlag = true;JogTimeCnt = 1;JogTime=millis();break;//if GO_BACK code is detected, then backward
    case STOP_STOP: stop_Stop();JogTime = 0;break;//stop
    default:break;
  }
  Drive_Num=DEF;
 //keep current moving mode for  200 millis seconds
  if(millis()-JogTime>=200)
  {
    JogTime=millis();
    if(JogFlag == true)
    {
      stopFlag = false;
      if(JogTimeCnt <= 0)
      {
        JogFlag = false; stopFlag = true;
      }
      JogTimeCnt--;
    }
```

```
    if(stopFlag == true)
    {
      JogTimeCnt=0;
      stop_Stop();
    }
   }
}

void setup()
{
  pinMode(RightDirectPin1, OUTPUT);
  pinMode(RightDirectPin2, OUTPUT);
  pinMode(speedPinL, OUTPUT);
  pinMode(LeftDirectPin1, OUTPUT);
  pinMode(LeftDirectPin2, OUTPUT);
  pinMode(speedPinR, OUTPUT);
  stop_Stop();

  pinMode(IR_PIN, INPUT);
  digitalWrite(IR_PIN, HIGH);
  IR.enableIRIn();
}


void loop()
{
  do_IR_Tick();
  do_Drive_Tick();
}
```