

/\*

MINISUMO\_GENESIS2\_NONESTARTMODULE\_CORE400\_TRIMPOT\_TACTICS\_0127  
MINI SUMO M1 || REM ROBOT version0127

From ROBOTUS & JSUMO 2019/08

Address: Hilal Rifatpaşa Mah., Perpa İş Ticaret Merkezi, B Blok, 8.  
Kat, No. 1094, ŞİŞLİ, İSTANBUL

Telephone: +90 (212) 222 07 19

www.robotus.net && www.jsumo.com

Written by ENG. Abdul Raouf DEHNI

Designed by ENG. Firat Fares DEDE

MainBoard : GENESIS ver2

Sensors : Opponent Sensors : MR45 \* 3

Line Sensors : ML1 || QTR1A \* 2

Motors : JSUMO 6V 400RPM \* 2

Wheels : SLT20 \* 2

Battery : LI-PO 3S 11.1V 450mAh

\*/

/\*

STANDARD CODE NAMES FOR ROBOTUS &JSUMO 2019/08

LEFT SENSOR = LSEN

LEFT FRONT SENSOR = LFSEN

MIDDLE SENSOR = MSEN

RIGHT FRONT SENSOR = RFSEN

RIGHT SENSOR = RSEN

LEFT LINE SENSOR = LLINE

RIGHT LINE SENSOR = RLINE

MIDDLE OR BACK LINE SENSOR = MLINE

ULTRASONIC TRIG = TRIG

ULTRASONIC ECHO = ECHO

LEFT MOTOR PWM = LPWM

LEFT MOTOR DIRECTION = LDIR

LEFT MOTOR 1 = L1

LEFT MOTOR 2 = L2

RIGHT MOTOR PWM = RPWM

```

RIGHT MOTOR DIRECTION = RDIR
RIGHT MOTOR 1 = R1
RIGHT MOTOR 2 = R2

DIPSWITCH1 = DS1
DIPSWITCH2 = DS2
DIPSWITCH3 = DS3

START BUTON = BTN

TURN TRIMPOT = TMP //default Trimpot
SPEED TRIMPOT = STP

BUZZER = BZR

LEFT LED OR FIRST LED = LLED //default Led
MIDDLE LED = MLED
RIGHT LED = RLED
*/
// SENSORS
#define LLINE A0
#define RLINE A1
#define LSEN A2
#define MSEN A3
#define RSEN A4
//MOTORS
#define LPWM 3
#define LDIR 12
#define RPWM 11
#define RDIR 13
//SWITCHS, BUTONS, TRIMPOTS ,BUZZER &LEDS
#define DS1 5
#define DS2 6
#define DS3 7
#define BTN 10
#define SPD A7
#define TRN A6
#define BZR 9
#define LLED 8
#define LSS 300 //LINE SENSOR SENSIBILITY
//VARIABLES
int LastValue = 3;
int SPEED, LCM, RCM;

```

```
bool TACTIC = HIGH;

void setup() {
  //INPUTS
  pinMode(LLINE, INPUT);
  pinMode(RLINE, INPUT);
  pinMode(LSEN, INPUT);
  pinMode(MSEN, INPUT);
  pinMode(RSEN, INPUT);
  pinMode(DS1, INPUT);
  pinMode(DS2, INPUT);
  pinMode(DS3, INPUT);
  pinMode(BTN, INPUT);
  pinMode(SPD, INPUT);
  pinMode(TRN, INPUT);
  //OUTPUTS
  pinMode(LDIR, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(RDIR, OUTPUT);
  pinMode(RPWM, OUTPUT);
  pinMode(BZR, OUTPUT);
  pinMode(LLED, OUTPUT);
  //INITIAL VALUES
  digitalWrite(LLINE, HIGH);
  digitalWrite(RLINE, HIGH);
  digitalWrite(LSEN, HIGH);
  digitalWrite(MSEN, HIGH);
  digitalWrite(RSEN, HIGH);

  digitalWrite(LPWM, LOW);
  digitalWrite(LDIR, LOW);
  digitalWrite(RPWM, LOW);
  digitalWrite(RDIR, LOW);

  digitalWrite(DS1, HIGH);
  digitalWrite(DS2, HIGH);
  digitalWrite(DS3, HIGH);
  digitalWrite(BTN, HIGH);
  digitalWrite(BZR, LOW);
  digitalWrite(LLED, LOW);

  digitalWrite(LLED, HIGH);
  tone(BZR, 500);
```

```

    delay(300);
    digitalWrite(LLED, LOW);
    tone(BZR, 250);
    delay(700);
    noTone(BZR);
    Serial.begin(9600);
}

void MOVE( float Lval, float Rval, int timex) {
    Lval = Lval * 2.55;
    Rval = Rval * 2.55;
    if ( Lval >= 0 ) {
        digitalWrite(LDIR, HIGH);
    }
    else {
        Lval = abs(Lval);
        digitalWrite(LDIR, LOW);
    }
    analogWrite(LPWM, Lval);
    if ( Rval >= 0 ) {
        digitalWrite(RDIR, HIGH);
    }
    else {
        Rval = abs(Rval);
        digitalWrite(RDIR, LOW);
    }
    analogWrite(RPWM, Rval);
    delay(timex);
}

void DEBUG() {
    SPEED = float((1023 - analogRead(SPD)) / 34.1); //MAXIMUM 30%
    LCM = float(analogRead(TRN) / 102.3); // MAXIMUM 10%
    RCM = 10 - LCM; // MAXIMUM 10%
    Serial.print(" LineSensors:");
    Serial.print(" LLINE:");
    Serial.print(analogRead(LLINE));
    Serial.print(" RLINE:");
    Serial.print(analogRead(RLINE));
    Serial.print(" OpponentSensors:");
    Serial.print(" LSEN:");
    Serial.print(digitalRead(LSEN));
    Serial.print(" MSEN:");
}

```

```

Serial.print(digitalRead(MSEN));
Serial.print(" RSEN:");
Serial.print(digitalRead(RSEN));
Serial.print("  DIPSWITCHS:");
Serial.print(" DS1:");
Serial.print(digitalRead(DS1));
Serial.print(" DS2:");
Serial.print(digitalRead(DS2));
Serial.print(" DS3:");
Serial.print(digitalRead(DS3));
Serial.print("  BUTONS:");
Serial.print(" BTN:");
Serial.print(digitalRead(BTN));
Serial.print("  TRIMPOTS:");
Serial.print(" SPD:");
Serial.print(analogRead(SPD));
Serial.print(" SPEED:");
Serial.print(SPEED);
Serial.print(" LCM:");
Serial.print(LCM);
Serial.print(" RCM:");
Serial.print(RCM);
Serial.print(" TRN:");
Serial.println(analogRead(TRN));
delay(100);
}

void loop() {
SLEEPMODE:
  DEBUG();
  if (digitalRead(LLINE) == LOW || digitalRead(LSEN) == LOW ||
digitalRead(MSEN) == LOW || digitalRead(RSEN) == LOW ||
digitalRead(RLINE) == LOW)
    digitalWrite(LLED, HIGH);
  else
    digitalWrite(LLED, LOW);
  if (digitalRead(BTN) == LOW) {
    while (digitalRead(BTN) == LOW);
    for (int i = 0 ; i < 5 && digitalRead(BTN) == HIGH; i++) {
      digitalWrite(LLED, HIGH);
      tone(BZR, 500);
      delay(500);
      if (digitalRead(BTN) == LOW) {

```

```

    noTone(BZR);
    digitalWrite(LLED, LOW);
    while (digitalRead(BTN) == LOW);
    goto SLEEPMODE;
}
digitalWrite(LLED, LOW);
noTone(BZR);
delay(500);
if (digitalRead(BTN) == LOW) {
    while (digitalRead(BTN) == LOW);
    goto SLEEPMODE;
}
}
if (TACTIC == HIGH) {
    TACTIC = LOW;
    if (digitalRead(DS1) == LOW && digitalRead(DS2) == LOW &&
digitalRead(DS3) == LOW) { //111
        goto START;
    }
    else if (digitalRead(DS1) == HIGH && digitalRead(DS2) == HIGH &&
digitalRead(DS3) == HIGH) { //000
        goto START;
    }
    else if (digitalRead(DS1) == HIGH && digitalRead(DS2) == LOW &&
digitalRead(DS3) == HIGH) { //010
        MOVE(100, 100, 80);
    }
    else if (digitalRead(DS1) == HIGH && digitalRead(DS2) == LOW &&
digitalRead(DS3) == LOW) { //011 ZIGZAG RIGHT TO LEFT
        MOVE( 100, -100, 70);
        MOVE(-100, 100, 30);

        MOVE( 100, 100, 100);
        MOVE(-100, -100, 30);

        MOVE( -100, 100, 120);
        MOVE(100, -100, 30);

        MOVE( 100, 100, 200);
        MOVE(-100, -100, 30);
    }
    else if (digitalRead(DS1) == LOW && digitalRead(DS2) == LOW &&
digitalRead(DS3) == HIGH) { //110 ZIGZAG LEFT TO RIGHT

```

```

    MOVE(-100, 100, 70);
    MOVE( 100, -100, 30);

    MOVE( 100, 100, 100);
    MOVE(-100, -100, 30);

    MOVE( 100, -100, 120);
    MOVE(-100, 100, 30);

    MOVE( 100, 100, 200);
    MOVE(-100, -100, 30);
}
else if (digitalRead(DS1) == HIGH && digitalRead(DS2) == HIGH &&
digitalRead(DS3) == LOW) { //001 TO RIGHT
    MOVE(100, -100, 105);
    MOVE(-100, 100, 40);

    MOVE( 30, 30, 1);
}
else if (digitalRead(DS1) == LOW && digitalRead(DS2) == HIGH &&
digitalRead(DS3) == HIGH) { //100 TO LEFT
    MOVE(-100, 100, 105);
    MOVE(100, -100, 40);

    MOVE( 30, 30, 1);
}
else if (digitalRead(DS1) == LOW && digitalRead(DS2) == HIGH &&
digitalRead(DS3) == LOW) { //101 TO BEHIND
    MOVE(100, -100, 180);
    MOVE(-100, 100, 40);

    MOVE( 30, 30, 1);
}
}
}
START:
while (digitalRead(BTN) == HIGH)
{
    SPEED = float((1023 - analogRead(SPD)) / 34.1); //MAXIMUM 30%
    LCM = float((1023 - analogRead(TRN)) / 102.3); // MAXIMUM 10%
    RCM = float(analogRead(TRN) / 102.3); // MAXIMUM 10%
    if (analogRead(LLINE) < LSS && analogRead(RLINE) < LSS) {
        MOVE(-100, -100, 1);
        digitalWrite(LLED, HIGH);
    }
}
}

```

```

        while (analogRead(LLINE) < LSS && analogRead(RLINE) < LSS &&
digitalRead(BTN) == HIGH);
        MOVE(-100, -100, 30);
        MOVE(100, -100, 150);
        MOVE(-100, 100, 30);
        LastValue = 3;
    }
    else if (analogRead(LLINE) < LSS && analogRead(RLINE) > LSS) {
        MOVE(-100, -100, 1);
        digitalWrite(LLED, HIGH);
        while (analogRead(LLINE) < LSS && analogRead(RLINE) > LSS &&
digitalRead(BTN) == HIGH);
        MOVE(-100, -100, 30);
        MOVE(100, -100, 130);
        MOVE(-100, 100, 30);
        LastValue = 3;
    }
    else if (analogRead(LLINE) > LSS && analogRead(RLINE) < LSS) {
        MOVE(-100, -100, 1);
        digitalWrite(LLED, HIGH);
        while (analogRead(LLINE) > LSS && analogRead(RLINE) < LSS &&
digitalRead(BTN) == HIGH);
        MOVE(-100, -100, 30);
        MOVE(-100, 100, 130);
        MOVE(100, -100, 30);
        LastValue = 3;
    }
    else if (digitalRead(MSEN) == LOW) {
        MOVE(100, 100, 1);
        digitalWrite(LLED, HIGH);
        while (digitalRead(MSEN) == LOW && analogRead(LLINE) > LSS &&
analogRead(RLINE) > LSS && digitalRead(BTN) == HIGH);
        LastValue = 3;
    }
    else if (digitalRead(RSEN) == LOW) {
        MOVE(100, -100, 1);
        digitalWrite(LLED, HIGH);
        LastValue = 5;
    }
    else if (digitalRead(LSEN) == LOW) {
        MOVE(-100, 100, 1);
        digitalWrite(LLED, HIGH);
        LastValue = 1;
    }

```



```

}
else if (LastValue == 1) {
    MOVE((-SPEED - LCM), (SPEED + RCM), 1);
    digitalWrite(LLED, LOW);
}
else if (LastValue == 3) {
    MOVE((SPEED + LCM), (SPEED + RCM), 1); // (SPEED + _CM) = (30% +
10% ) = 40% MAXIMUM
    digitalWrite(LLED, LOW);
}
else if (LastValue == 5) {
    MOVE((SPEED + LCM), (-SPEED - RCM), 1);
    digitalWrite(LLED, LOW);
}
}
if (digitalRead(BTN) == LOW) {
    MOVE( 0, 0, 1);
    while (digitalRead(BTN) == LOW);
    LastValue = 3;
    TACTIC = HIGH;
    goto SLEEPMODE;
}
}
}

```