*Autonomous Car Running Version 2.0*

1. UltraSonic Sensor

 2. LineTracking Sensor

 3. Manual-Automatic mode shift button

 4. Connection to the Raspberry Pi

* UltraSonic ==> LineTracking
* LineTracking ==> Complete Running
* Button pressed ==> Switch mode
* Wait on input from Raspberry Pi ==> User able to control from GUI (to be finished in version 3.0)

“”””””””””””””””””””””””””””””””””””””””””””

<<Pseudo Code>>

int checkUltraSonic();

if gonna hit -> return 0

else             -> return 1

int checkLineTracking();

if it get off track   -> return 0

else                     -> return 1

if Button is pressed

Enter interrupt Mode

Establish Connection with the Raspberry Pi

Use UART protocol

“”””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””

<<Source Code>>

volatile int carMode ;      // Volatile in order to make sure data are stored properly during Interrupt

int raspberry\_pi\_input ;

int button               = 2   ;

int motorR1            = A0 ;

int motorR2            = A1 ;

int motorL1             = A2 ;

int motorL2              = A3 ;

int LTsensorR           =  3 ;

int LTsensorL            =  4 ;

int USsensorR\_trig    =  5 ;

int USsensorR\_echo  = 6 ;

int USsensorL\_trig    = 7 ;

int USsensorL\_echo =  8 ;

int USsensorM\_trig   = 9 ;

int USsensorM\_echo = 10 ;

int LTsensorR\_value  ;

int LTsensorL\_value  ;

int USsensorR\_value ;

int USsensorL\_value ;

int USsensorM\_value ;

         \*/ “””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””” \*/

int enterManualMode ();                    // To be implemented in version 3.0

int checkUltraSonic()

{

     digitalWrite(USsensorR\_trig, HIGH);

     delayMicroseconds(2) ;

     digitalWrite(USsensorR\_trig, LOW);

                                                                                // Evaluting the Right u.s. sensor

     digitalWrite(USsensorR\_trig, HIGH);

     delayMicroseconds(10) ;

     digitalWrite(USsensorR\_trig, LOW);

     USsensorR\_value = pulseIn(USsensorR\_echo , HIGH) \* 0.034 / 2 ;

     digitalWrite(USsensorL\_trig, HIGH);

     delayMicroseconds(2) ;

     digitalWrite(USsensorL\_trig, LOW);

                                                                                  // Evaluting the Left u.s. sensor

     digitalWrite(USsensorL\_trig, HIGH);

     delayMicroseconds(10) ;

     digitalWrite(USsensorL\_trig, LOW);

     USsensorL\_value = pulseIn(USsensorL\_echo , HIGH) \* 0.034 / 2 ;

     digitalWrite(USsensorM\_trig, HIGH);

     delayMicroseconds(2) ;

     digitalWrite(USsensorM\_trig, LOW);

                                                                                    // Evaluting the Middle u.s. sensor

     digitalWrite(USsensorM\_trig, HIGH);

     delayMicroseconds(10) ;

     digitalWrite(USsensorM\_trig, LOW);

     USsensorM\_value = pulseIn(USsensorM\_echo , HIGH) \* 0.034 / 2 ;

  if (USsensorR\_value >15 && USsensorL\_value >15  && USsensorM\_value >35 )  {return 1 ;}

  else                          {return 0 ;}

}

int checkLineTracking()

{

 LTsensorR\_value = digitalRead(3);

 LTsensorL\_value = digitalRead(4);

 if (LTsensorR\_value  && LTsensorL\_value == 1) { return 1 ;}

 else                                                                      { return 0 ;}

}

void moveForward ()

{

  digitalWrite(motorR1  , HIGH) ;

  digitalWrite(motorR2  ,LOW) ;

  digitalWrite(motorL1  , HIGH) ;

  digitalWrite(motorL2  , LOW) ;

}

void turnRight ()

{

  digitalWrite(motorR1  , LOW) ;

  digitalWrite(motorR2  ,LOW) ;

  digitalWrite(motorL1  , HIGH) ;

  digitalWrite(motorL2  , LOW) ;

}

void turnLeft ()

{

 digitalWrite(motorR1  , HIGH) ;

  digitalWrite(motorR2  ,LOW) ;

  digitalWrite(motorL1  , LOW) ;

  digitalWrite(motorL2  , LOW) ;

}

void stop()

{

  digitalWrite(motorR1  , LOW) ;

  digitalWrite(motorR2  , LOW) ;

  digitalWrite(motorL1  , LOW) ;

  digitalWrite(motorL2  , LOW) ;

}

void moveBack()

{

  digitalWrite(motorR1  , LOW) ;

  digitalWrite(motorR2  ,HIGH) ;

  digitalWrite(motorL1  , LOW) ;

  digitalWrite(motorL2  , HIGH) ;

}

void escapeAlgorithm1 ()             // Avoids near objects in the front

{

    stop() ;

    do { moveBack() ;}       while (! checkUltraSonic() ) ;

    turnLeft() ;

    delay (200) ;

    moveForward() ;

    delay(1500) ;

    turnRight() ;

    delay(200) ;

    moveForward() ;

    delay (2700) ;

    turnRight() ;

    delay(200) ;

    moveForward() ;

    delay(1500) ;

    turnLeft() ;

    delay (200) ;

    moveForward() ;

 }

void escapeAlgorithm2 ()          // Avoids far objects in the front

{

  do { turnRight() ;}     while (! checkUltraSonic() ) ;

  moveForward() ;

  delay(1500) ;

  turnLeft() ;

  delay(200) ;

  moveForward() ;

  delay (3500);

  turnLeft() ;

  delay(200) ;

  moveForward() ;

  delay(1500) ;

  turnRight()  ;

  delay(200) ;

  moveForward() ;

}

void escapeAlgorithm3 ()          // Avoids objects on the right

{

  do { turnLeft() ;}     while (! checkUltraSonic() ) ;

  turnRight() ;

  delay(100) ;

  moveForward() ;

  delay(1200);

  turnRight() ;

  delay(100) ;

  moveForward() ;

  delay(1200);

  turnLeft() ;

  delay(100) ;

  moveForward() ;

}

void escapeAlgorithm4 ()          // Avoids objects on the left

{

  do { turnRight() ;}     while (! checkUltraSonic() ) ;

  turnLeft() ;

  delay(100) ;

  moveForward() ;

  delay(1200);

  turnLeft() ;

  delay(100) ;

  moveForward() ;

  delay(1200);

  turnRight() ;

  delay(100) ;

  moveForward() ;

}

void switchCarMode ()         //  ISR function

{

   stop() ;

   carMode ++ ;

}

\*/ “””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””” /\*

void setup()

{

attachInterrupt(digitalPinToInterrupt(2), switchCarMode, CHANGE) ;    // The Interrupt

Serial.begin(9600);                      // For connection with Raspberry pi

pinMode(motorR1  ,OUTPUT) ;

pinMode(motorR2   ,OUTPUT) ;

pinMode(motorL1     ,OUTPUT)  ;

pinMode(motorL2  ,OUTPUT)  ;

pinMode(LTsensorR  ,INPUT) ;

pinMode(LTsensorL  ,INPUT) ;

pinMode(USsensorR\_trig  ,OUTPUT);

pinMode(USsensorR\_echo  ,INPUT) ;

pinMode(USsensorL\_trig  ,OUTPUT);

pinMode(USsensorL\_echo  ,INPUT) ;

pinMode(USsensorM\_trig  ,OUTPUT);

pinMode(USsensorM\_echo  ,INPUT) ;

}

\*/ ”””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””””” \*/

void loop()

{

   if (carMode %2) {

                              enterManualMode ();                    // To be implemented in version 3.0

                              }

while(checkUltraSonic()) {                                          // Automatic Mode

move forward() ;

      checkLineTracking() ;

if( LTsensorR\_value == 0) {

                                                      do { turnRight () ;}

                                                      while (! checkLineTracking()) ;

                                                       }

else if( LTsensorL\_value == 0) {

                                                   do { turnLeft () ;}

                                                   while (! checkLineTracking()) ;

                                                   }

else if( ! checkLineTracking()) {

                                                     do {

turnRght () ; moveForward() ; delay(1000) ; turnLeft()

moveForward() ; delay(1000) ;

}

while (! checkLineTracking()) ;

                                                     }

}

 if ( USsensorM\_value =< 20) {

                                                  escapeAlgorithm1 () ;

                                                 }

else if ( USsensorM\_value =< 35 && USsensorM\_value  > 20) {

                                                                                                 escapeAlgorithm2 () ;

                                                                                                 }

else  if ( USsensorR\_value =< 20  ) {

                                                     escapeAlgorithm3 () ;

                                                       }

else  if ( USsensorL\_value =< 20  ) {

                                                     escapeAlgorithm4 () ;

                                                      }

   if(Serial.available() > 0) {              // Connection with the Raspberry Pi using UART protocol

     raspberry\_pi\_input = Serial.read() ;

   }

}