Group 4 - David Wants Lolis Team(DWLT)

Our progress(from hackmd)

11/16

- The car can connect to both server and cellphone, and it’s remoteable.
- Finish the “Start” for car starts to go, “Done” for car stops

11/23

- Confirm that we have 5 members in our group.

11/27

- Car can move now(via server, without phone)
- Need small adjustments.
- Came out with first prototype code, Can get to the end but without obstacle in the middle.

11/28

- Trying to use ultrasonic sensor to detect the obstacle, but fail with strange return value.
  (for example: 4cm return 4cm, 5cm return 5cm, 6cm return 6cm or 25cm by random).
    - Cause by broken ultrasonic sensor? <-- No
    - Cause by Serial.begin? <-- Change to 9600, still strange numbers
    - (Guess)Cause by wifi and ultrasonic share a same thread, then ultrasonic need to wait wifi signal return, so the value goes wrong.
- Try second prototype code.
- Finish the Final code by second prototype.

11/29

- Test everyone’s car, check if it is stable.
- Look like Divik’s 7697 has issue connecting to the server, his car on the map will be unstable to be monitored.
- The unstable of the car may caused by server and people walking around when testing.

Code description
(Only describe important algorithm. Other function or code is from TA’s code or basic function.)

Create a task to ask position from server

```
xTaskCreate(
        askPos,        /* Task function. */
        "askPos",      /* String with name of task. */
        10000,         /* Stack size in words. */
        NULL,          /* Parameter passed as input of the task */
        1,              /* Priority of the task. */
        NULL
);          /* Task handle. */
```

However, this function will let us delay our ultrasound return value(since it needs to finish askPos then we can finfally get ultrasound return value) as we described at11/28 in hackmd.
Therefore this time we didn’t use ultrasound module.
The method is we move over the half point of map(obstacle) then start to use vector direction to make car turn to the final point.

AskPos function

```
void askPos( void * parameter )
{
   while(1){
      if ((messageLen = wifiClient.available()) > 0) {
         int i = 0;
         do{
            buf[i++] = wifiClient.read();
         }while(i<32 && buf[i-1]!='\r');
         buf[i-1] = '\0';
         recv_ID = strtok(buf,"|\0");
         recv_buf = strtok(NULL,"|\0");
         if(strcmp(recv_buf, "Start") == 0){
            timetogo = true;
            detection = 0;
         }
         else if(strcmp(recv_buf, "Done") == 0){
            timetogo = false;
            detection = 0;
         }
         else
            sscanf(recv_buf,"POS:(%d,%d)(%d,%d)",&MyPosX,&MyPosY,&DstPosX,&DstPosY);
         Serial.println(MyPosX);
         Serial.println(MyPosY);
         send_mes("Position","");
      }
   }
   vTaskDelete( NULL );

}

void loop()
{
```

Get initial x and y position

```
   if(timetogo == true){   //for game start
      static const int InitPosX = MyPosX;
      static const int InitPosY = MyPosY;
```

2 step of our vector calculation,in this step(step 1) we start to run over half of the map.
Because car is pretty hard to turn back if the car’s direction turns too far, we need to adjust car on the way.

```
    if(detection == 0){
      forward(0);
      if(abs(InitPosX - MyPosX) > 240 || abs(InitPosY - MyPosY) > 200){
        freeze(100);
        static const int MidDirX = MyPosX - InitPosX;
        static const int MidDirY = MyPosY - InitPosY;
        static const double MidDir = atan2(MidDirY, MidDirX);
        static const int InitDirX = DstPosX - InitPosX;
        static const int InitDirY = DstPosY - InitPosY;
        static const double InitDir = atan2(InitDirY, InitDirX);
        if(MidDir - InitDir < 0 || MidDir - InitDir > PI)
          right(250);
        else if(MidDir - InitDir > 0)
          left(250);
        detection = 1;
      }
    }
```

2 step of our vector calculation, in this step (step2) we already run over half of the map, start to use current x and y position to move forward to the finish point.

```
    if(detection == 1){
      static int MidPosX = MyPosX;
      static int MidPosY = MyPosY;
      static const int EndDirX = DstPosX - MidPosX;
      static const int EndDirY = DstPosY - MidPosY;
      static const double EndDir = atan2(EndDirY, EndDirX);
      forward(0);
      delay(50);
      double MyDir = atan2(MyPosY - MidPosY, MyPosX - MidPosX);
      if(MyDir - EndDir < 0 || MyDir - EndDir > PI){
        slightly_right(75);
        forward(50);
      }
      else if(MyDir - EndDir > 0){
        slightly_left(75);
        forward(50);
      }
    }
  }
```

Game end signal

```
  else if(timetogo == false){     //for game end
    freeze(0);
  }
```

```
}
```

Review

We run pretty good at the last week, but why it didn’t successfully run to the finish points this week?
We assume it’s because our battery is not fully charged.
Since the output voltage will affect the car velocity, all of us did not recharge the battery after the test last week.
So we can’t sure the remaining of the battery. This lead to the fail this week.