

Teil 4

Lektion

2

**Projektkurs:
Gierige Schlange**

Überblick:

In diesem Tutorial werden wir das Max7219-Anzeigemodul und das MPU6050-Modul verwenden, um ein "gieriges Schlangenspiel" zur Steuerung der Schwerkraft zu entwickeln. Durch Kippen des Boards von einer Seite zur anderen können wir die Bewegung der Schlange steuern und sie findet das Essen.

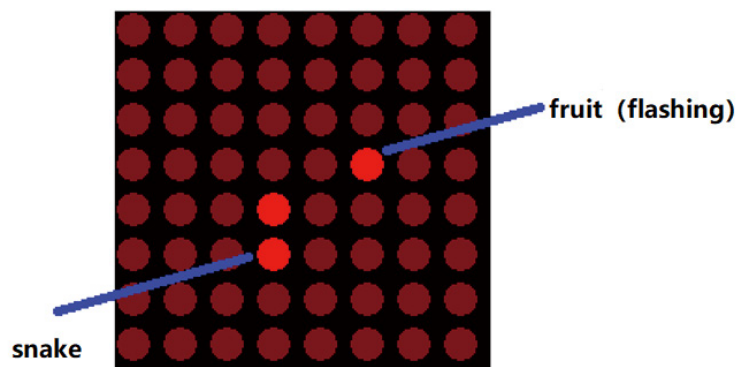
Benötigte Komponenten:

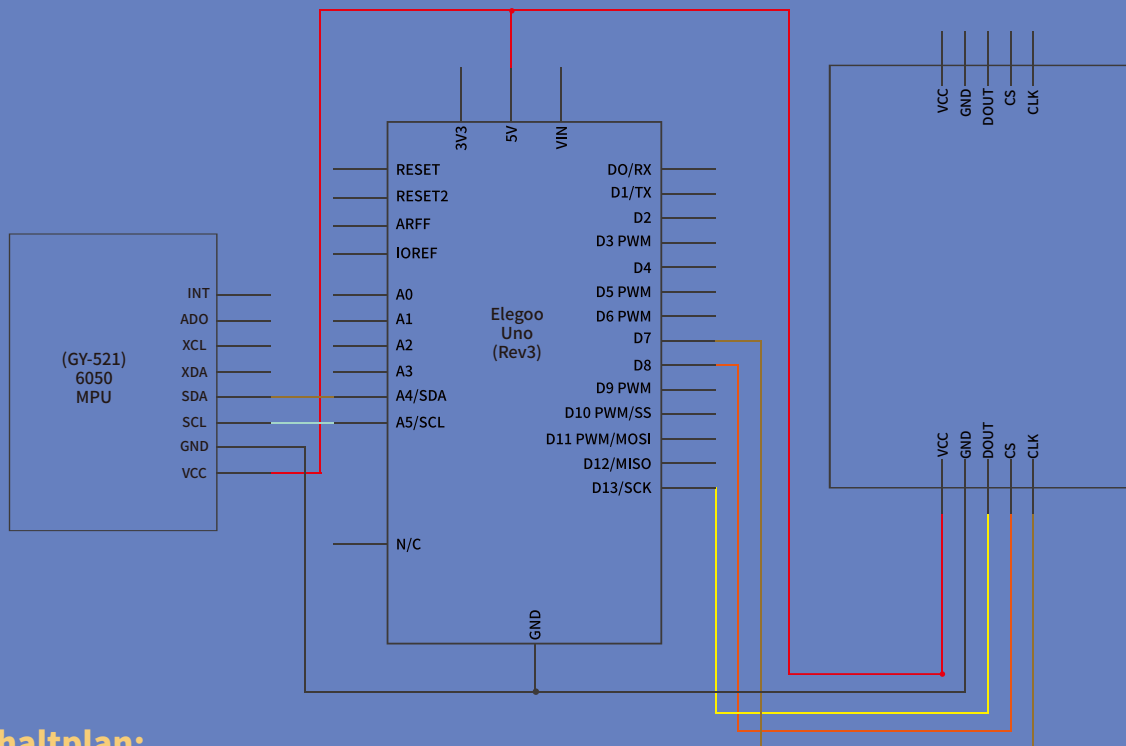
- (1) X Elegoo UNO R3
- (1) X GY521
- (1) X MAX7219
- (1) X ALL IN ONE Sensor Shield



Spiel-Einführung:

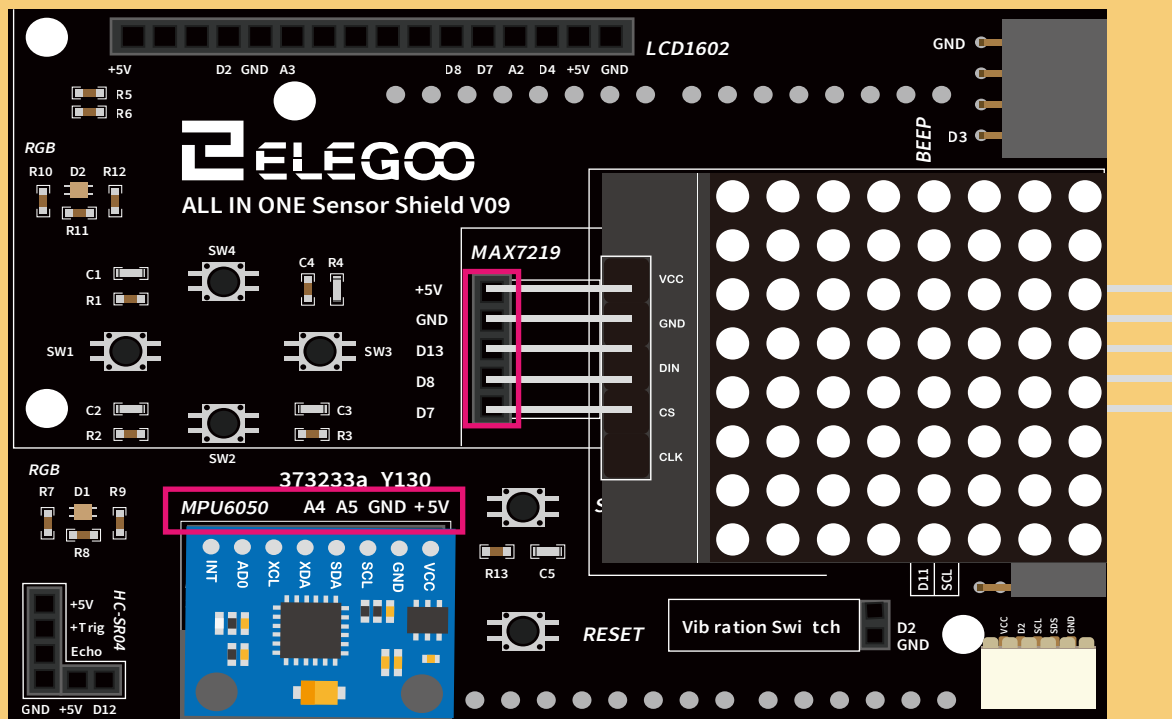
Die Schlange bewegt sich laufend weiter und die Schwerkraft erfasst die Richtung des Fortschritts der Schlange. Wenn die Schlange das Essen frisst, wächst der Schlangenkörper und die Schlange bewegt sich schneller. Das Spiel endet, wenn die Schlange sich selbst berührt.





Schaltplan:



Tips: Bitte fügen Sie das erweiterte Board in die UNO ein.



Verbindungs-Übersicht:

Code:

Stellen Sie sicher, dass Sie diese Bibliotheken hinzufügen, bevor Sie das Programm

 LedControl.zip	2019/8/14 16:06	9 KB
 Wire.zip	2019/8/14 16:10	17 KB

Part1:

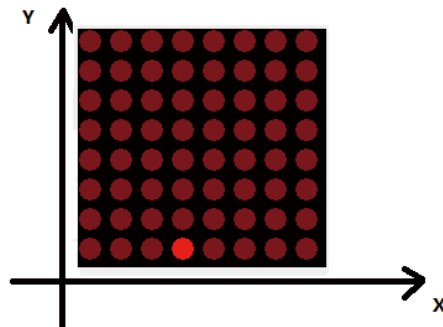
Kontrollieren wir die Bewegung der Schlange

1. Die Darstellung der Schlange auf dem Display

```
int snakeX[64];      // X-coordinates of snake
int snakeY[64];      // Y-coordinates of snake
snakeX[0] = 0;
snakeY[0] = 4;
int snakeLength = 2;
void drawSnake(){
  for(int i=0; i<snakeLength; i++){
    /* Set the status of a single Led.
    * Params :
    * addr address of the display
    * row  the row of the Led (0..7)
    * col  the column of the Led (0..7)
    * state If true the led is switched on, if false it is switched off

    setLed(int addr, int row, int col, boolean state);
    */
    lc.setLed(0, snakeX[i], snakeY[i], true);
  }
}
```

Erstellen Sie zwei Arrays, um die Position der Schlange in der 8 * 8-Punktmatrix darzustellen:
snakeX [0], snakeY [0] indicates the position of the first section of the snake.
snakeX[1], snakeY[1] indicates the position of the second section of the snake.
snakeX[2], snakeY[2] indicates the position of the third section of the snake.



2. Die Bewegungs-Kontrolle

Zuerst werden die Knoten der Schlange durch die for-Schleife um eins nach hinten verschoben.

Zum Beispiel: snakeLength = 3: Der dritte Teil der Schlange wird durch den zweiten Teil der Schlange ersetzt. Der zweite Teil der Schlange wird durch den ersten Teil der Schlange ersetzt. Der erste Teil ist nach der for-Schleife leer.

Verwenden Sie dann die switch-Funktion, um die Richtung des ersten Teils auszuwählen.

3. Holen Sie sich den Befehl für die nächste Richtung der Schlange

Sie ändern die Richtung, wenn sich der GY521 in verschiedenen Richtungen bewegt. Gehen Sie dann in Schritt 2 zur Switch-Funktion, um die Änderung der Richtung abzuschließen.

```
void nextstep(){
  for(int i=snakeLength-1; i>0; i--){
    snakeX[i] = snakeX[i-1];
    snakeY[i] = snakeY[i-1];
  }

  switch(direction){
    case UP:
      snakeY[0] = snakeY[0]-1;
      if(snakeY[0]==-1)
        snakeY[0]=7;
      break;
    case RIGHT:
      snakeX[0] = snakeX[0]+1;
      if(snakeX[0]==8)
        snakeX[0]=0;
      break;
    case DOWN:
      snakeY[0] = snakeY[0]+1;
      if(snakeY[0]==8)
        snakeY[0]=0;
      break;
    case LEFT:
      snakeX[0]=snakeX[0]-1;
      if(snakeX[0]==-1)
        snakeX[0]=7;
      break;
  }
}
```

```
void checkGy521(){
  static unsigned long delaytime = millis();
  if(millis() - delaytime > 20){
    delaytime = millis();
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
    AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
    delay(33);
    Serial.print(AcX);
    Serial.print('\t');
    Serial.println(AcY);
    if(AcX>-1000&&AcX<1000&&AcY>8000&&direction!= DOWN){
      direction=UP;
      Serial.println("DOWN");
    }
    else if(AcX>-1000&&AcX<1000&&AcY<-8000&& direction!= UP){
      direction=DOWN;
      Serial.println("UP");
    }
    else if(AcX<-8000&& AcY>-1000&&AcY<1000&& direction!= LEFT){
      direction=RIGHT;
      Serial.println("LEFT");
    }
    else if(AcX>8000&&AcY>-1000&&AcY<1000&& direction!= RIGHT){
      direction=LEFT;
      Serial.println("RIGHT");
    }
  }
}
```

Part2

Zufälliges Essen

1.Essen auf dem Display darstellen

Die Anzeige von Essen auf dem Display verläuft analog zur Darstellung der Schlange. Wenn Sie jedoch die Zeit in der Funktion beurteilen, blinkt das Essen. Dies ist ein Unterschieds-Kriterium zur Schlange.

```
void drawFruit(){
  if(inPlayField(fruitX, fruitY)){
    unsigned long currenttime = millis();
    if(currenttime - fruitPrevTime >= fruitBlinkTime){
      /*
       *Fruit keeps flashing
       */
      fruitLed = (fruitLed == true) ? false : true;
      fruitPrevTime = currenttime;
    }
    lc.setLed(0,fruitX, fruitY, fruitLed);
  }
}
```

2.zufälliges Essen

Verwenden Sie die Zufallsfunktion, um die Position des Essens zufällig zu erhalten. Das Essen soll nicht auf der Schlange selbst erscheinen.

```
void makeFruit(){
  int x, y;
  x = random(0, 8);
  y = random(0, 8);
  while(isPartOfSnake(x, y)){
    x = random(0, 8);
    y = random(0, 8);
  }
  fruitX = x;
  fruitY = y;
}
```

Part3

Schlange isst das Essen

Wenn sich die Schlange bewegt und das Essen frisst, so nimmt ihre Länge zu und mit zunehmender Länge nimmt auch die Bewegungs-Geschwindigkeit zu.

```
if((snakeX[0] == fruitX) && (snakeY[0] == fruitY)){
  snakeLength++;
  score++;
  if(snakeLength < MAX_SNAKE_LENGTH){
    makeFruit();
  }
  else {
    fruitX = fruitY = -1;
  }
  if(score%8==0)
  {
    delayTime = delayTime - 50;
  }
}
```

Part4

Die Schlange berührt sich selbst

■ Beurteilungsbedingung: Überprüfen Sie, ob der erste Knoten der Schlange mit dem anderen Knoten der Schlange übereinstimmt. Wenn sich die Schlange selbst trifft, starte das Spiel neu.

```
void snakeItSelf(){ // check if snake touches itself
    for(int i=1;i<snakeLength;i++){
        if((snakeX[0] == snakeX[i] && (snakeY[0] == snakeY[i])){
            gameOver();
        }
    }
}

void gameOver(){
    lc.clearDisplay(0);
    for(int r = 0; r < 8; r++){
        for(int c = 0; c < 8; c++){
            lc.setLed(0, r, c, HIGH);
            delay(50);
        }delay(50);
    }
    delay(300);
    score = 3;
    snakeLength = 3;
    direction = RIGHT;
    snakeX[0]=3;
    snakeY[0]=4;
    delayTime = 500;
    loop();
}
```