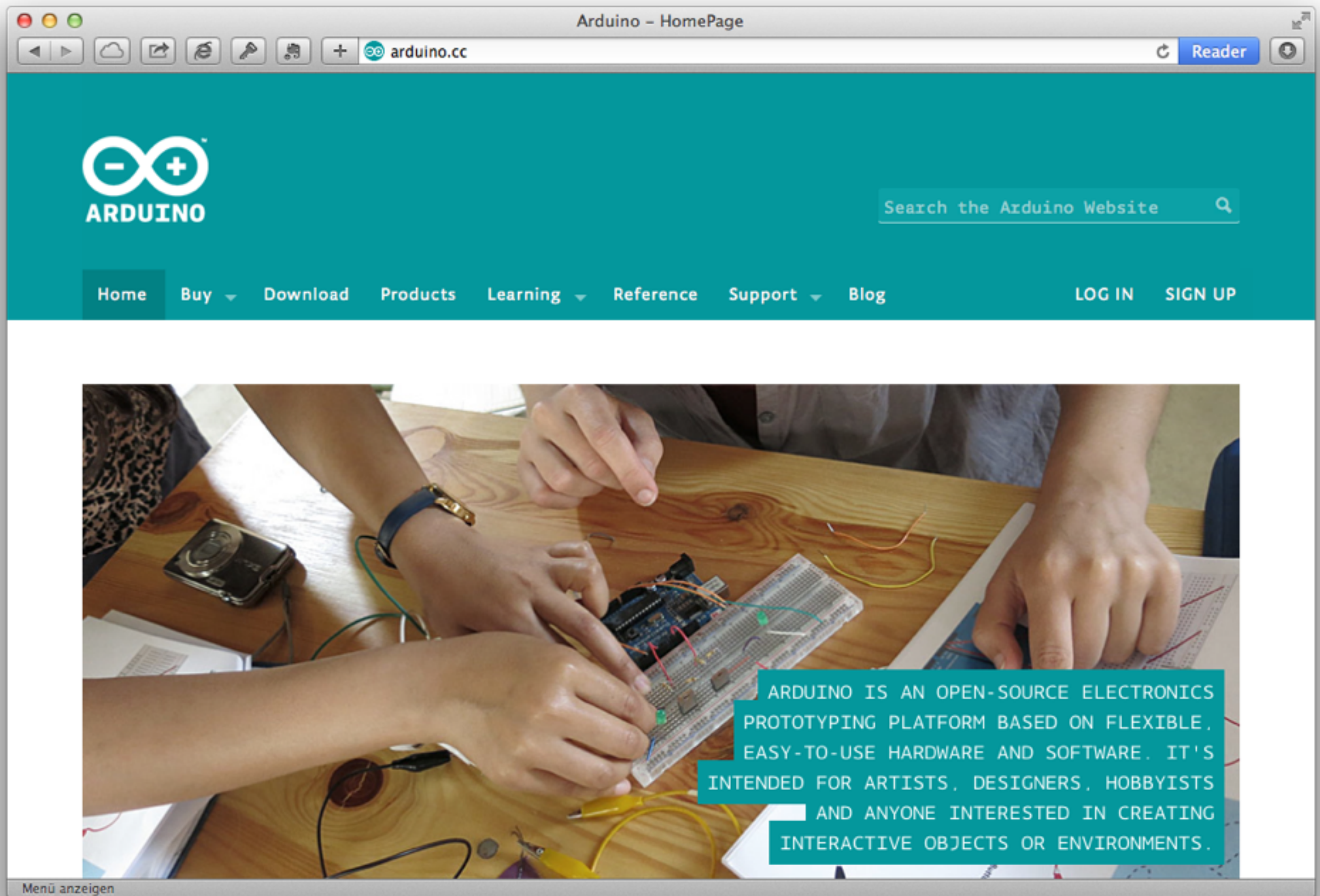


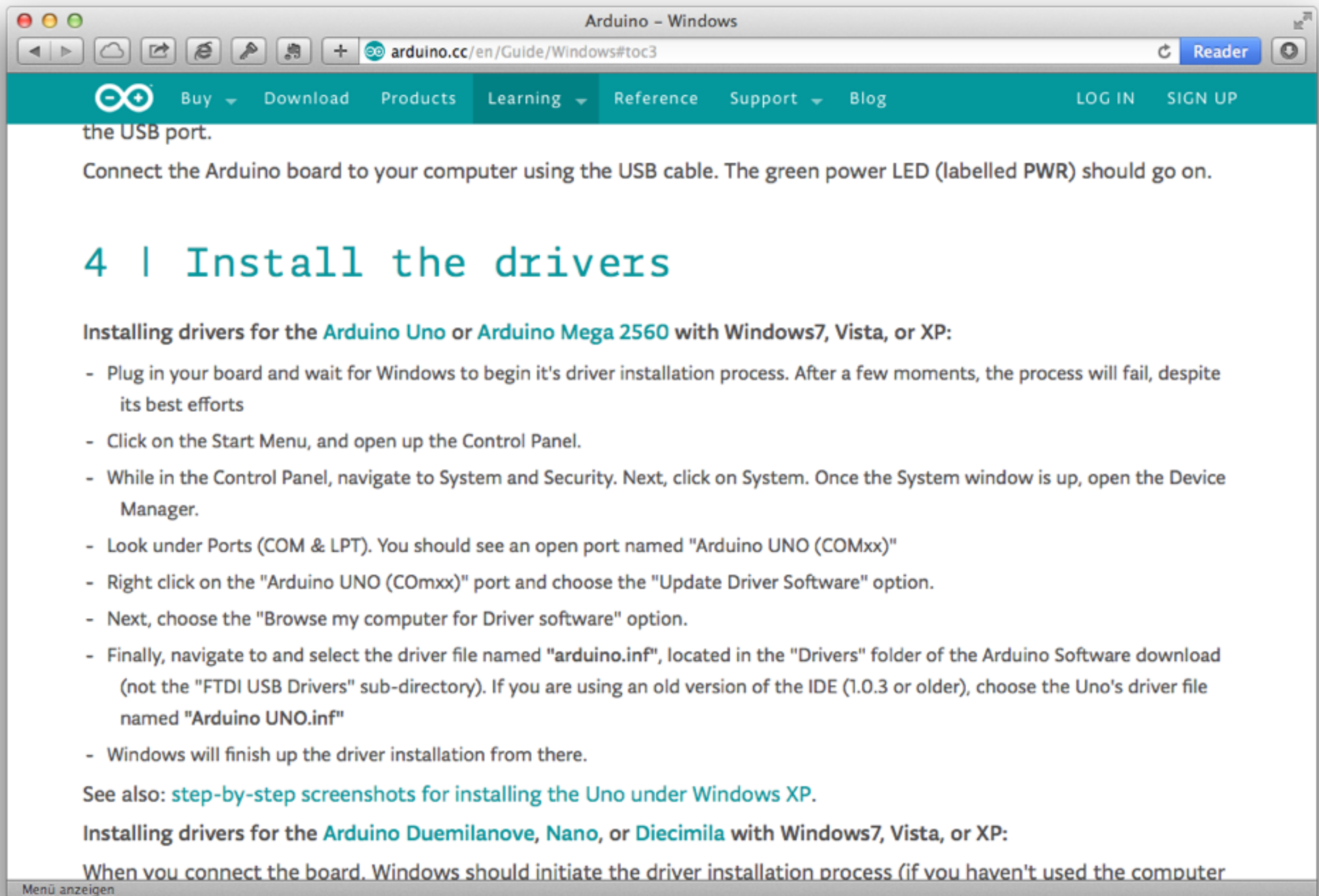
# DramaLab

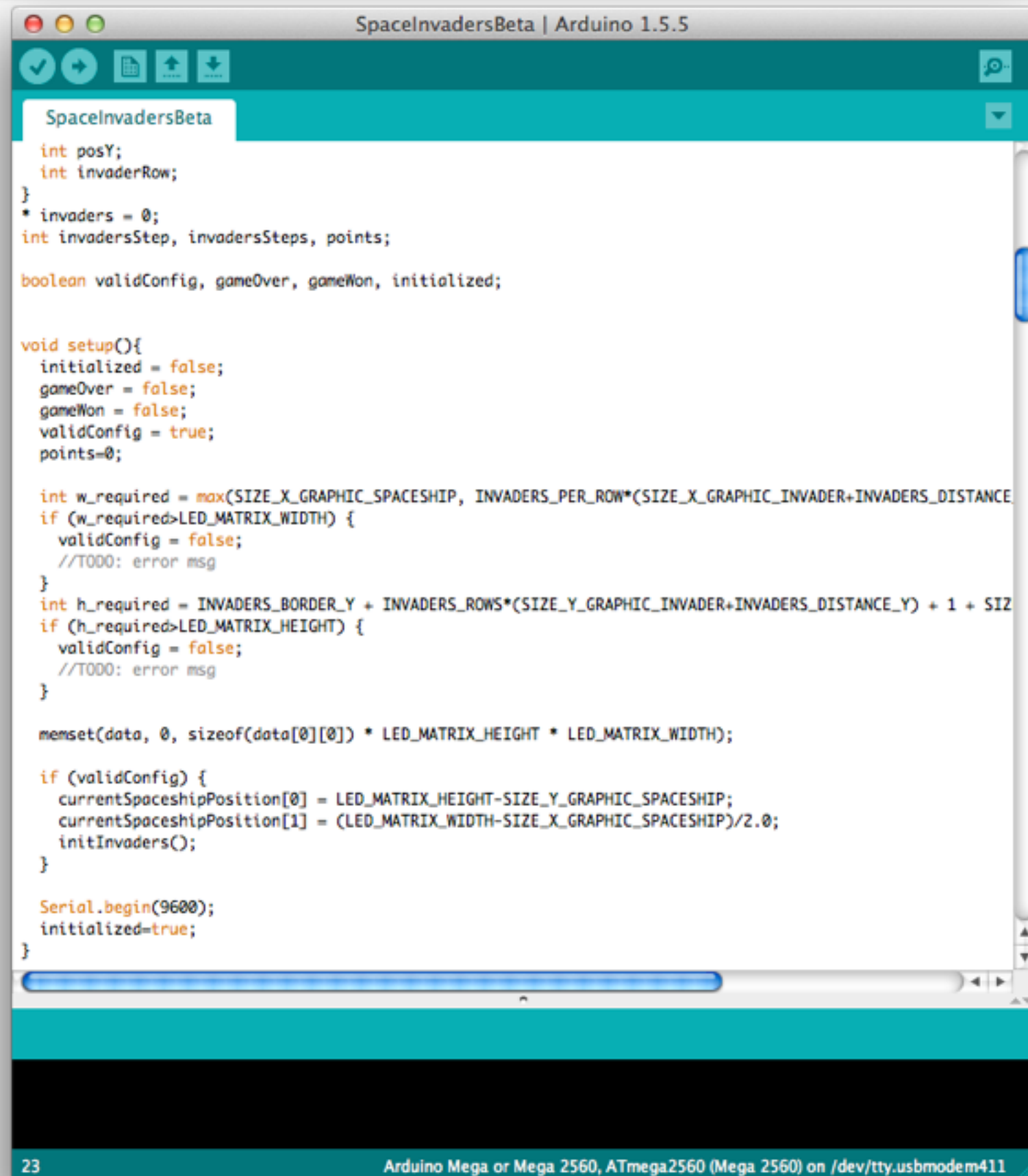
## Arduino IDE











```

SpaceInvadersBeta
int posY;
int invaderRow;
}
* invaders = 0;
int invadersStep, invadersSteps, points;

boolean validConfig, gameOver, gameWon, initialized;

void setup(){
  initialized = false;
  gameOver = false;
  gameWon = false;
  validConfig = true;
  points=0;

  int w_required = max(SIZE_X_GRAPHIC_SPACESHIP, INVADERS_PER_ROW*(SIZE_X_GRAPHIC_INVADER+INVADERS_DISTANCE);
  if (w_required>LED_MATRIX_WIDTH) {
    validConfig = false;
    //TODO: error msg
  }
  int h_required = INVADERS_BORDER_Y + INVADERS_ROWS*(SIZE_Y_GRAPHIC_INVADER+INVADERS_DISTANCE_Y) + 1 + SIZE_Y_GRAPHIC_SPACESHIP;
  if (h_required>LED_MATRIX_HEIGHT) {
    validConfig = false;
    //TODO: error msg
  }

  memset(data, 0, sizeof(data[0][0]) * LED_MATRIX_HEIGHT * LED_MATRIX_WIDTH);

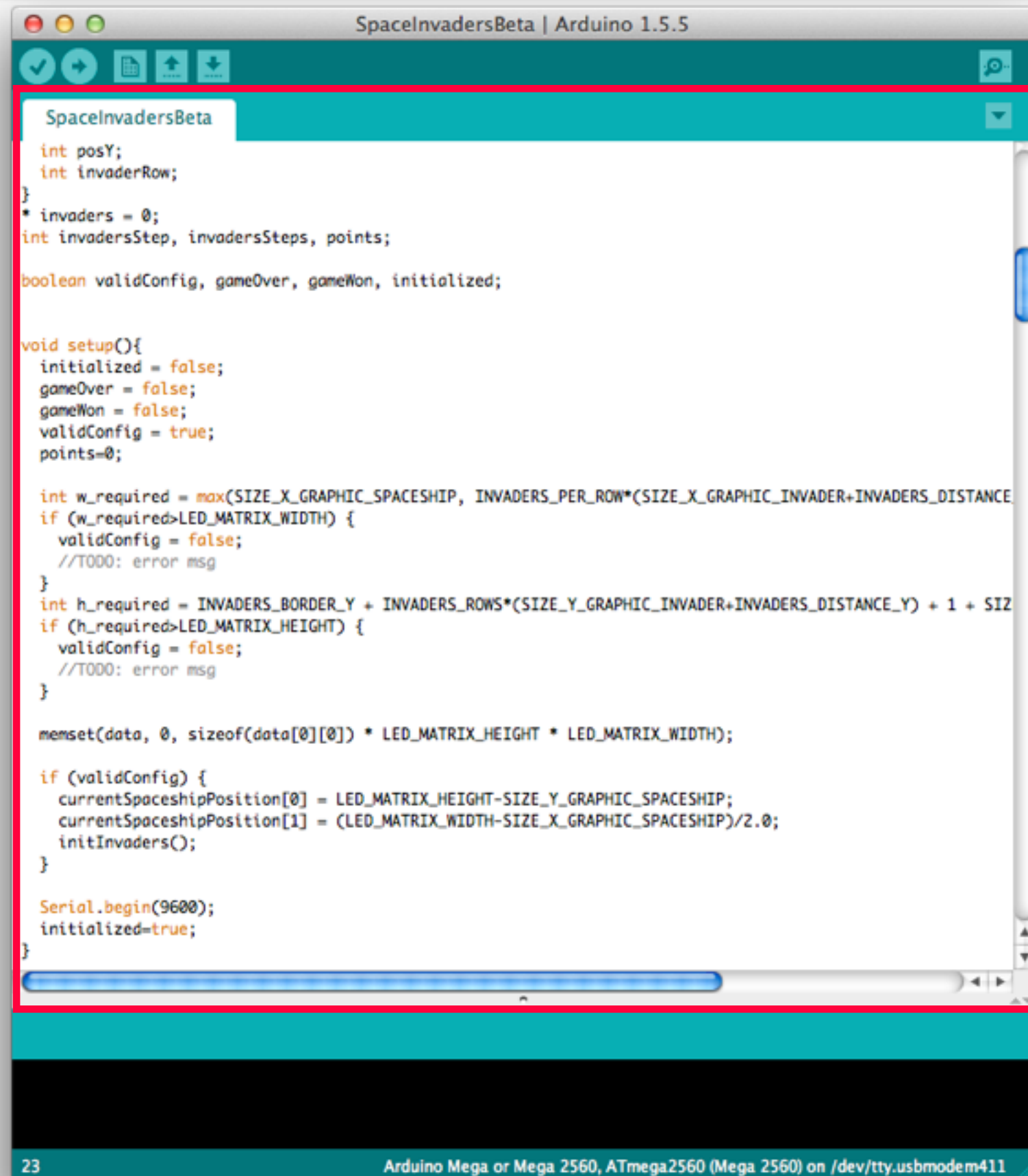
  if (validConfig) {
    currentSpaceshipPosition[0] = LED_MATRIX_HEIGHT-SIZE_Y_GRAPHIC_SPACESHIP;
    currentSpaceshipPosition[1] = (LED_MATRIX_WIDTH-SIZE_X_GRAPHIC_SPACESHIP)/2.0;
    initInvaders();
  }

  Serial.begin(9600);
  initialized=true;
}

```

23 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411

## Code Window



The screenshot shows the Arduino IDE interface with the 'SpaceInvadersBeta' sketch open in the Code Window. The sketch is written in C++ and includes variables for game state, setup functions, and matrix dimensions. The Code Window is highlighted with a red border. The status bar at the bottom indicates the board is 'Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411'.

```
SpaceInvadersBeta
int posY;
int invaderRow;
}
* invaders = 0;
int invadersStep, invadersSteps, points;

boolean validConfig, gameOver, gameWon, initialized;

void setup(){
  initialized = false;
  gameOver = false;
  gameWon = false;
  validConfig = true;
  points=0;

  int w_required = max(SIZE_X_GRAPHIC_SPACESHIP, INVADERS_PER_ROW*(SIZE_X_GRAPHIC_INVADER+INVADERS_DISTANCE
  if (w_required>LED_MATRIX_WIDTH) {
    validConfig = false;
    //TODO: error msg
  }
  int h_required = INVADERS_BORDER_Y + INVADERS_ROWS*(SIZE_Y_GRAPHIC_INVADER+INVADERS_DISTANCE_Y) + 1 + SIZ
  if (h_required>LED_MATRIX_HEIGHT) {
    validConfig = false;
    //TODO: error msg
  }

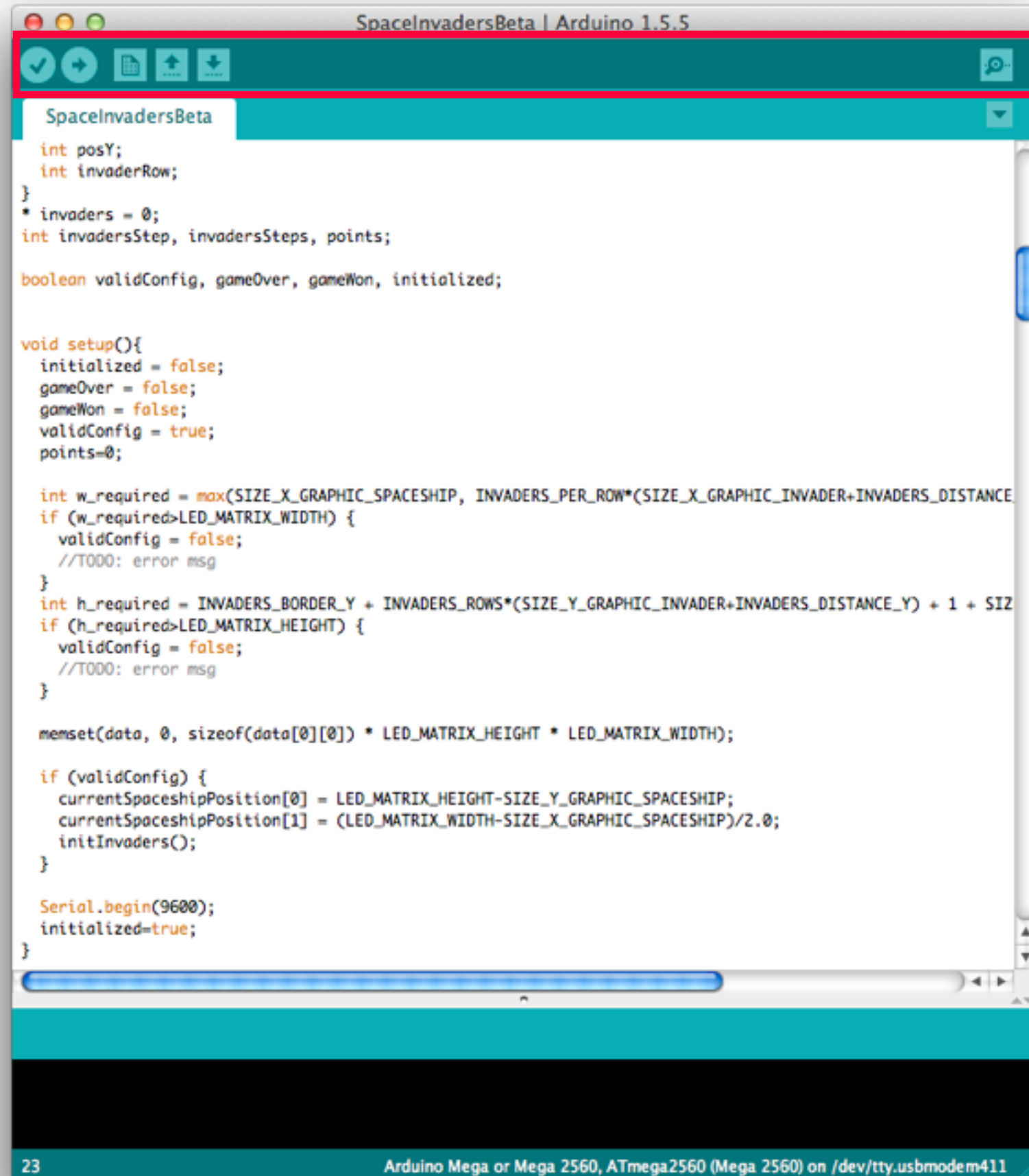
  memset(data, 0, sizeof(data[0][0]) * LED_MATRIX_HEIGHT * LED_MATRIX_WIDTH);

  if (validConfig) {
    currentSpaceshipPosition[0] = LED_MATRIX_HEIGHT-SIZE_Y_GRAPHIC_SPACESHIP;
    currentSpaceshipPosition[1] = (LED_MATRIX_WIDTH-SIZE_X_GRAPHIC_SPACESHIP)/2.0;
    initInvaders();
  }

  Serial.begin(9600);
  initialized=true;
}

23 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411
```

Verify  
Upload  
New  
Open  
Save  
Serial Monitor



```
SpaceInvadersBeta | Arduino 1.5.5

int posY;
int invaderRow;
}
* invaders = 0;
int invadersStep, invadersSteps, points;

boolean validConfig, gameOver, gameWon, initialized;

void setup(){
  initialized = false;
  gameOver = false;
  gameWon = false;
  validConfig = true;
  points=0;

  int w_required = max(SIZE_X_GRAPHIC_SPACESHIP, INVADERS_PER_ROW*(SIZE_X_GRAPHIC_INVADER+INVADERS_DISTANCE
  if (w_required>LED_MATRIX_WIDTH) {
    validConfig = false;
    //TODO: error msg
  }
  int h_required = INVADERS_BORDER_Y + INVADERS_ROWS*(SIZE_Y_GRAPHIC_INVADER+INVADERS_DISTANCE_Y) + 1 + SIZ
  if (h_required>LED_MATRIX_HEIGHT) {
    validConfig = false;
    //TODO: error msg
  }

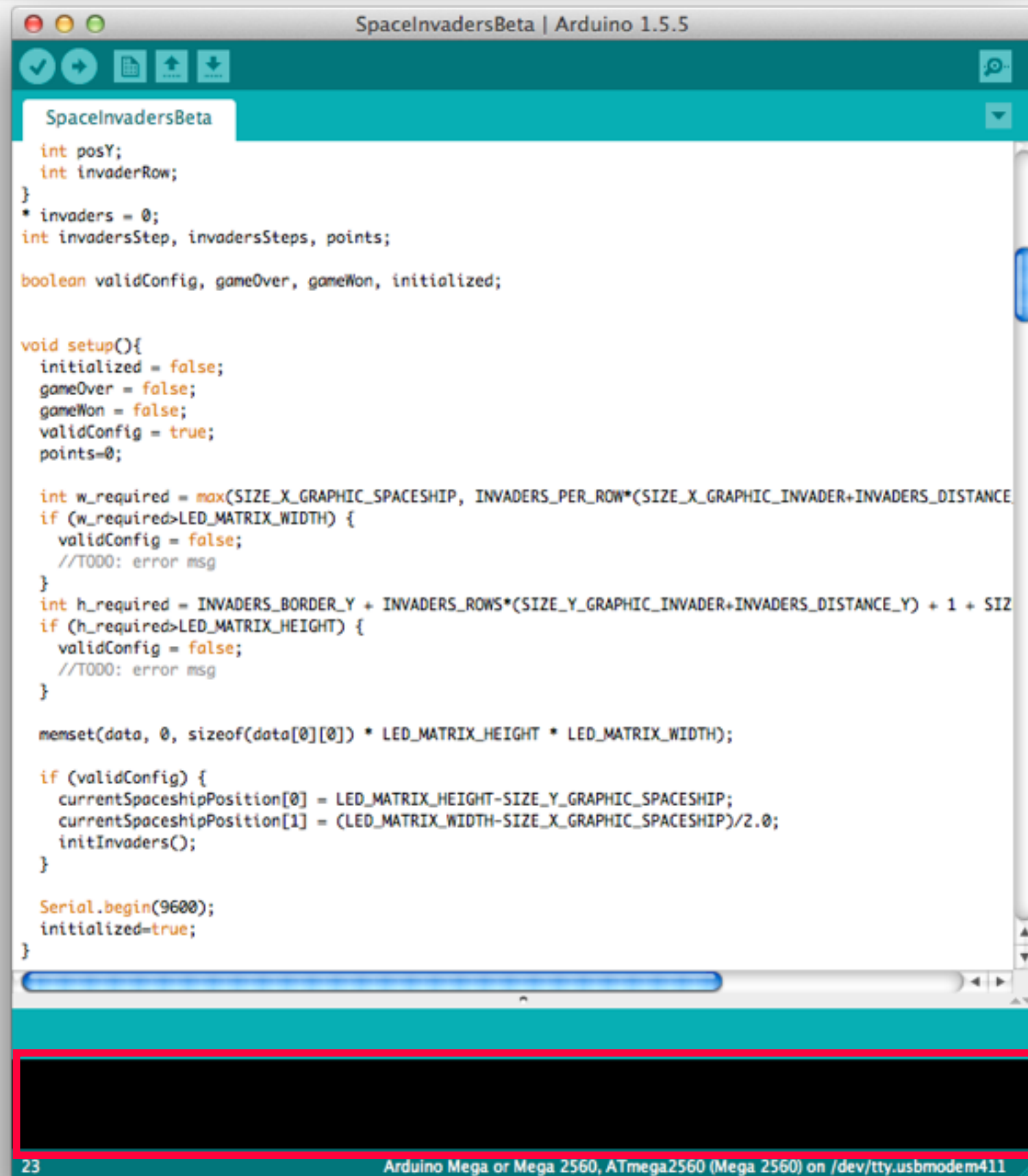
  memset(data, 0, sizeof(data[0][0]) * LED_MATRIX_HEIGHT * LED_MATRIX_WIDTH);

  if (validConfig) {
    currentSpaceshipPosition[0] = LED_MATRIX_HEIGHT-SIZE_Y_GRAPHIC_SPACESHIP;
    currentSpaceshipPosition[1] = (LED_MATRIX_WIDTH-SIZE_X_GRAPHIC_SPACESHIP)/2.0;
    initInvaders();
  }

  Serial.begin(9600);
  initialized=true;
}

23 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411
```





```
SpaceInvadersBeta
int posY;
int invaderRow;
}
* invaders = 0;
int invadersStep, invadersSteps, points;

boolean validConfig, gameOver, gameWon, initialized;

void setup(){
  initialized = false;
  gameOver = false;
  gameWon = false;
  validConfig = true;
  points=0;

  int w_required = max(SIZE_X_GRAPHIC_SPACESHIP, INVADERS_PER_ROW*(SIZE_X_GRAPHIC_INVADER+INVADERS_DISTANCE);
  if (w_required>LED_MATRIX_WIDTH) {
    validConfig = false;
    //TODO: error msg
  }
  int h_required = INVADERS_BORDER_Y + INVADERS_ROWS*(SIZE_Y_GRAPHIC_INVADER+INVADERS_DISTANCE_Y) + 1 + SIZE_Y_GRAPHIC_SPACESHIP;
  if (h_required>LED_MATRIX_HEIGHT) {
    validConfig = false;
    //TODO: error msg
  }

  memset(data, 0, sizeof(data[0][0]) * LED_MATRIX_HEIGHT * LED_MATRIX_WIDTH);

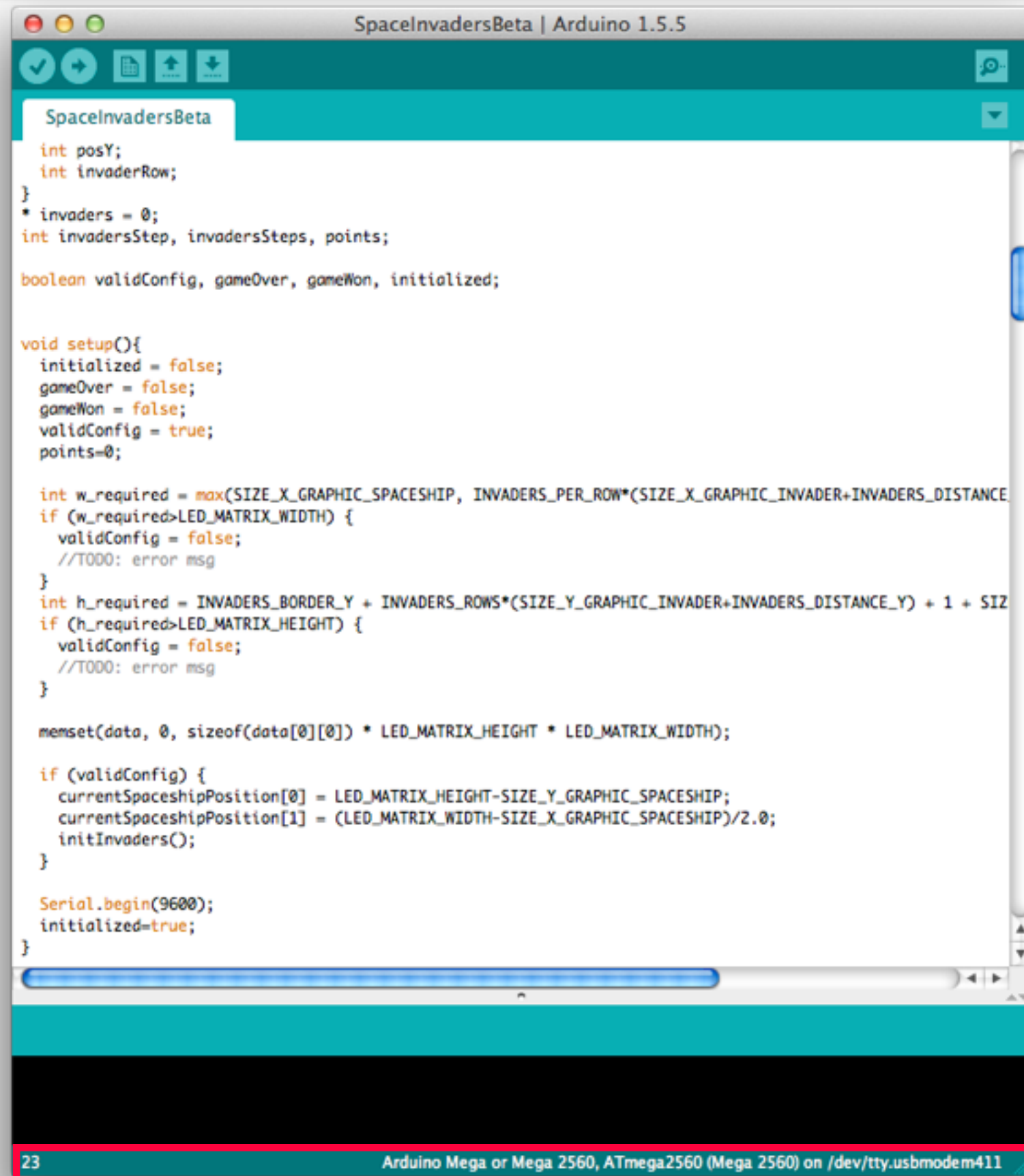
  if (validConfig) {
    currentSpaceshipPosition[0] = LED_MATRIX_HEIGHT-SIZE_Y_GRAPHIC_SPACESHIP;
    currentSpaceshipPosition[1] = (LED_MATRIX_WIDTH-SIZE_X_GRAPHIC_SPACESHIP)/2.0;
    initInvaders();
  }

  Serial.begin(9600);
  initialized=true;
}
```

23 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411

Debug  
Output





```
SpaceInvadersBeta | Arduino 1.5.5

SpaceInvadersBeta
int posY;
int invaderRow;
}
* invaders = 0;
int invadersStep, invadersSteps, points;

boolean validConfig, gameOver, gameWon, initialized;

void setup(){
  initialized = false;
  gameOver = false;
  gameWon = false;
  validConfig = true;
  points=0;

  int w_required = max(SIZE_X_GRAPHIC_SPACESHIP, INVADERS_PER_ROW*(SIZE_X_GRAPHIC_INVADER+INVADERS_DISTANCE);
  if (w_required>LED_MATRIX_WIDTH) {
    validConfig = false;
    //TODO: error msg
  }
  int h_required = INVADERS_BORDER_Y + INVADERS_ROWS*(SIZE_Y_GRAPHIC_INVADER+INVADERS_DISTANCE_Y) + 1 + SIZE_Y_GRAPHIC_SPACESHIP;
  if (h_required>LED_MATRIX_HEIGHT) {
    validConfig = false;
    //TODO: error msg
  }

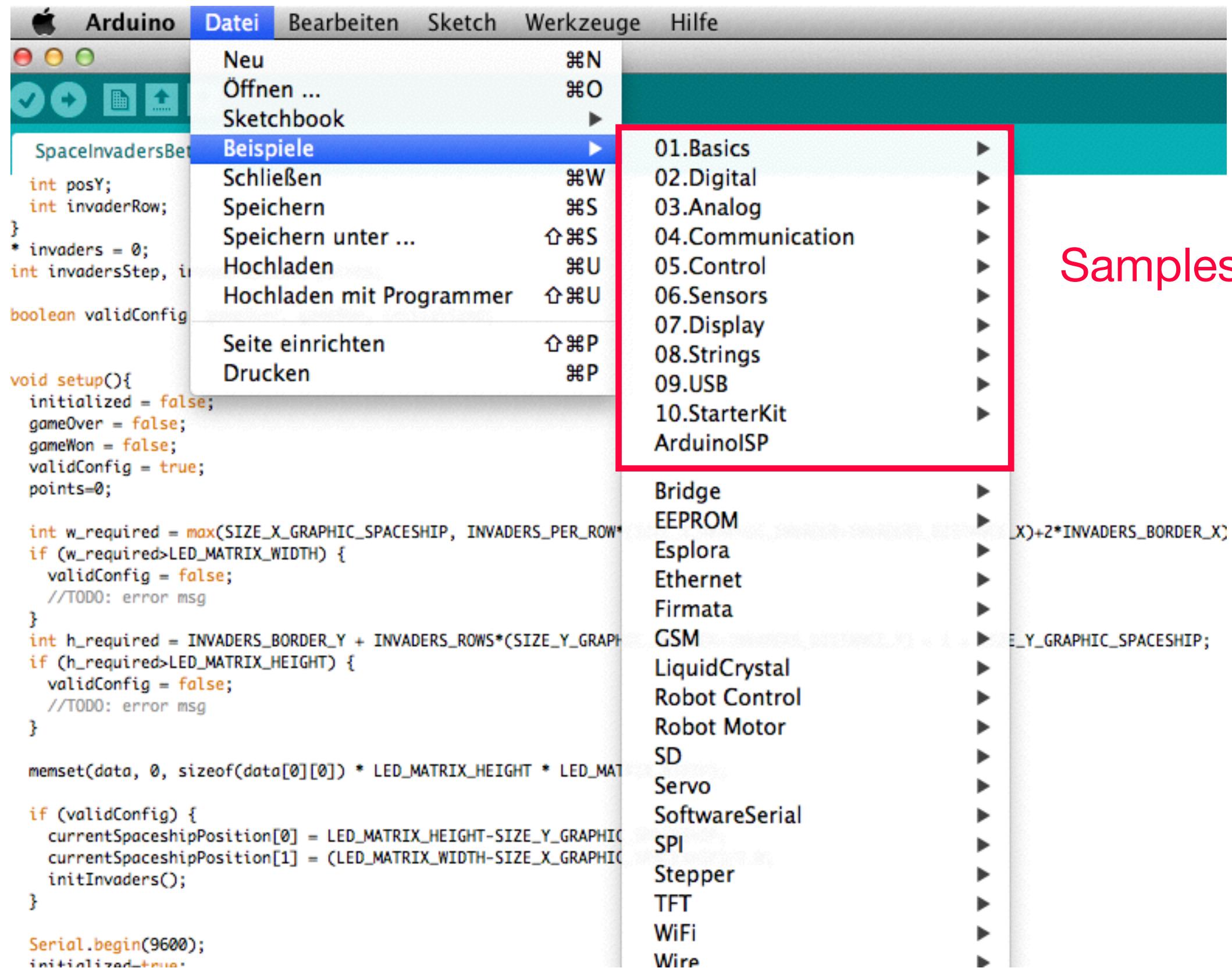
  memset(data, 0, sizeof(data[0][0]) * LED_MATRIX_HEIGHT * LED_MATRIX_WIDTH);

  if (validConfig) {
    currentSpaceshipPosition[0] = LED_MATRIX_HEIGHT-SIZE_Y_GRAPHIC_SPACESHIP;
    currentSpaceshipPosition[1] = (LED_MATRIX_WIDTH-SIZE_X_GRAPHIC_SPACESHIP)/2.0;
    initInvaders();
  }

  Serial.begin(9600);
  initialized=true;
}

23 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/tty.usbmodem411
```

Infos



Samples

