**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Направление 15.03.06 «Мехатроника и робототехника»

Отделение автоматизации и робототехники

Лабораторная работа №1 “ Знакомство с SPI. Применение библиотек при разработке ПО под STM32”

Наименование лабораторной работы

Отчет по лабораторной работе № 1

по дисциплине «Встраиваемые системы»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент гр. 8Е11 | \_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_ | И.С.Бондарев |
|  | Подпись | Дата | И.О. Фамилия |
| Проверил преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Н.И.Поберезкин |
| Должность | Подпись | Дата | И.О. Фамилия |

Томск – 2023

**Содержание**

[Цель работы 3](#_Toc1)

[Формулировка задания 3](#_Toc2)

[STM32CubeMX 4](#_Toc3)

[Блок-схема 6](#_Toc4)

[Код программы 9](#_Toc5)

[STMStudio 15](#_Toc6)

[Результат работы 17](#_Toc7)

[Выводы по работе 20](#_Toc8)

Цель работы

Научиться использовать последовательный синхронный стандарт передачи данных в режим полного дуплекса (SPI) и изучить строение LEDP10.

## 

Формулировка задания

Реализовать имитацию рабочего стола персонального компьютера используя светодиодную матрицу LED10, стандарт передачи данных SPI и джойстик.

STM32CubeMX

Производим первичную настойку платы в среду STM32CubeMX.



Рисунок 1 - Плата STM32

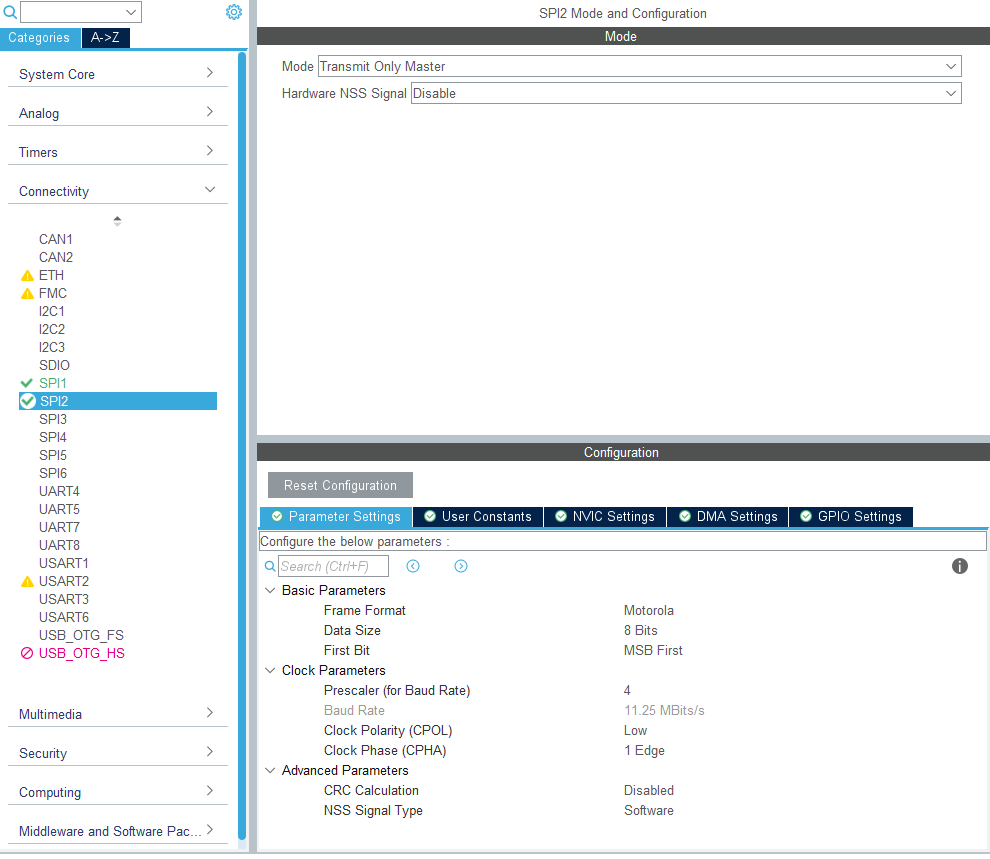


Рисунок 2 – Настройка SPI

Блок-схема

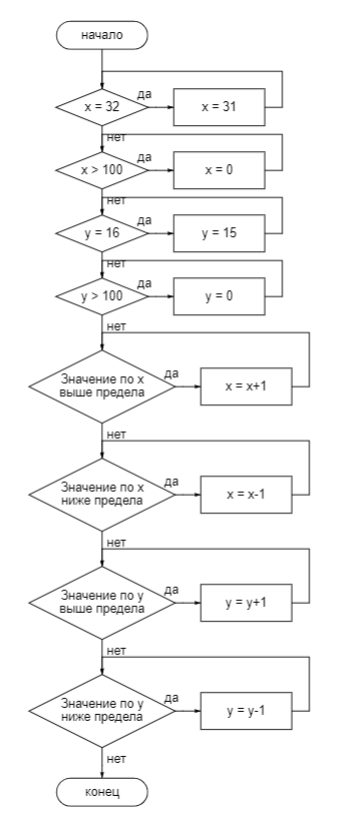


Рисунок 3 - Блок-схема корректировки координат текущего положения мыши

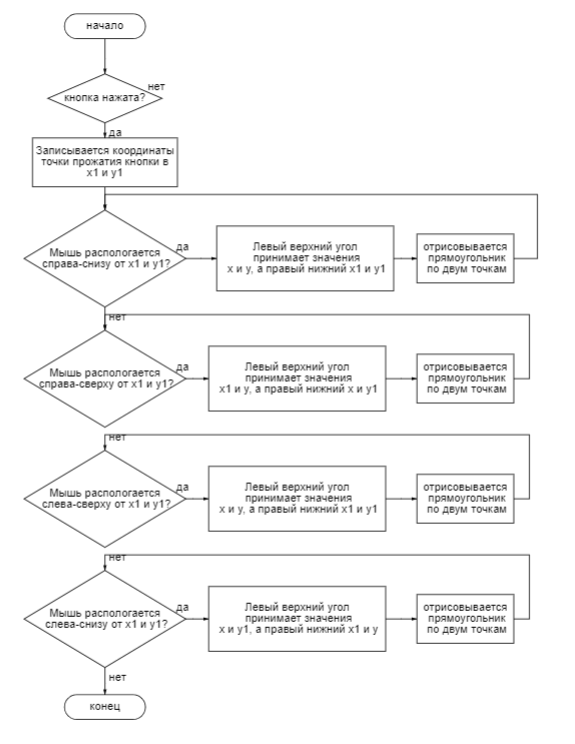


Рисунок 5 - Блок-схема отрисовки выделяемого поля

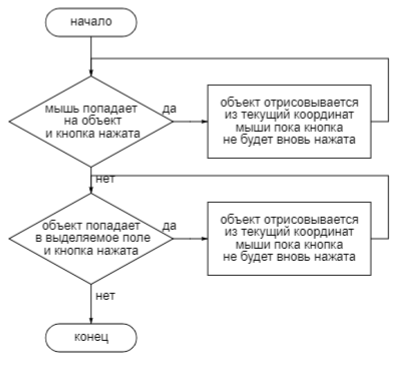


Рисунок 5 - Блок-схема переноса отрисованного объекта

Код программы

Листинг 1 - Программный код работы светодиодной панели

|  |
| --- |
| while (1)  {{  buttonstate = HAL\_GPIO\_ReadPin(button1\_GPIO\_Port, button1\_Pin);  HAL\_ADC\_Start\_IT(&hadc1);  HAL\_ADC\_Start\_IT(&hadc2);  HAL\_ADC\_PollForConversion(&hadc1, 10);  HAL\_ADC\_PollForConversion(&hadc2, 10);  AdcX = HAL\_ADC\_GetValue(&hadc1);  AdcY = HAL\_ADC\_GetValue(&hadc2);  pix[0].x = a;  pix[0].y = b;  objleftUp.x = m;  objleftUp.y = n;  objrightBottom.x = m1;  objrightBottom.y = n1;  cubeleftUp.x = p;  cubeleftUp.y = o;  cuberightBottom.x = p1;  cuberightBottom.y = o1;  if(flag == 1)  {  m = a;  n = b;  m1 = a+1;  n1 = b-1;  }  if(b < b1 && a > a1)  {  pixleftUp.x = a1;  pixleftUp.y = b1;  pixrightBottom.x = a;  pixrightBottom.y = b;  if(a1 < m && b1 > n && a > m1 && b < n1 && buttonstate == 0)  {  flag = 1;  }  }  if(b > b1 && a > a1)  {  pixleftUp.x = a1;  pixleftUp.y = b;  pixrightBottom.x = a;  pixrightBottom.y = b1;  if(a1 < m && b > n && a > m1 && b1 < n1 && buttonstate == 0)  {  flag = 1;  }  }  if(b > b1 && a < a1)  {  pixleftUp.x = a;  pixleftUp.y = b;  pixrightBottom.x = a1;  pixrightBottom.y = b1;  if(a < m && b > n && a1 > m1 && b1 < n1 && buttonstate == 0)  {  flag = 1;  }  }  if(b < b1 && a < a1)  {  pixleftUp.x = a;  pixleftUp.y = b1;  pixrightBottom.x = a1;  pixrightBottom.y = b;  if(a < m && b1 > n && a1 > m1 && b < n1 && buttonstate == 0)  {  flag = 1;  }  }  delay1++;  if(buttonstate == 0 && button == 0 && z == 0)  { if(delay1 > 2000)  {  button=1;  a1 = a;  b1 = b;  delay1 = 0;  }}  if(buttonstate == 1 && button == 0)  { a1 = a;  b1 = b; }  if(buttonstate == 1 && button == 1 && z == 0)  { if(delay1 > 2000)  { z = 1;  delay1 = 0; }}  if(buttonstate == 0 && button == 1 && z == 1)  { if(delay1 > 2000)  { button=0;  a1 = a;  b1 = b;  z = 0;  delay1 = 0;}}  if(buttonstate == 0 && ((a == m && b == n) || (a == m1 && b == n1) || (a == m1 && b == (n1 + 1)) || (a == (m1 - 1) && b == n1)) && flag == 0)  {button = 0;  a1 = a;  b1 = b;  flag = 1;  HAL\_Delay(500);}  if(buttonstate == 1 && ((a == m && b == n) || (a == m1 && b == n1) || (a == m1 && b == (n1 + 1)) || (a == (m1 - 1) && b == n1)) && flag == 1 && flag1 ==0)  {flag1 = 1;  button = 0;  a1 = a;  b1 = b;  HAL\_Delay(200); }  if(buttonstate == 0 && ((a == m && b == n) || (a == m1 && b == n1) || (a == m1 && b == (n1 + 1)) || (a == (m1 - 1) && b == n1)) && flag1 == 1)  { flag = 0;  flag1 = 0;  button = 0;  a1 = a;  b1 = b;  HAL\_Delay(200);}  delay2++;  clearScreen((uint8\_t\*)data);  if(button == 1)  {  drawRect(pixleftUp, pixrightBottom, (uint8\_t\*)data, P10AB\_11);  }  if(flag == 1)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_11);  drawRect(cubeleftUp, cuberightBottom, (uint8\_t\*)data, P10AB\_11);  }  if(flag == 0)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_11);  }  drawPixels((Pixel\*)pix, 1, (uint8\_t\*)data, P10AB\_11);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, A\_Channel\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, B\_Channel\_Pin, GPIO\_PIN\_SET);  HAL\_SPI\_Transmit(&hspi2, (uint8\_t\*)data, 16, 10);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_SET);  clearScreen((uint8\_t\*)data);  if(button == 1)  {  drawRect(pixleftUp, pixrightBottom, (uint8\_t\*)data, P10AB\_10);  }  if(flag == 1)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_10);  drawRect(cubeleftUp, cuberightBottom, (uint8\_t\*)data, P10AB\_10);  }  if(flag == 0)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_10);  }  drawPixels((Pixel\*)pix, 1, (uint8\_t\*)data, P10AB\_10);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, A\_Channel\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, B\_Channel\_Pin, GPIO\_PIN\_SET);  HAL\_SPI\_Transmit(&hspi2, (uint8\_t\*)data, 16, 10);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_SET);  clearScreen((uint8\_t\*)data);  if(button == 1)  {  drawRect(pixleftUp, pixrightBottom, (uint8\_t\*)data, P10AB\_01);  }  if(flag == 1)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_01);  drawRect(cubeleftUp, cuberightBottom, (uint8\_t\*)data, P10AB\_01);  }  if(flag == 0)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_01);  }  drawPixels((Pixel\*)pix, 1, (uint8\_t\*)data, P10AB\_01);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, A\_Channel\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, B\_Channel\_Pin, GPIO\_PIN\_RESET);  HAL\_SPI\_Transmit(&hspi2, (uint8\_t\*)data, 16, 10);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_SET);  clearScreen((uint8\_t\*)data);  if(button == 1)  {  drawRect(pixleftUp, pixrightBottom, (uint8\_t\*)data, P10AB\_00);  }  if(flag == 1)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_00);  drawRect(cubeleftUp, cuberightBottom, (uint8\_t\*)data, P10AB\_00);  }  if(flag == 0)  {  drawRect(objleftUp, objrightBottom, (uint8\_t\*)data, P10AB\_00);  }  drawPixels((Pixel\*)pix, 1, (uint8\_t\*)data, P10AB\_00);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, A\_Channel\_Pin, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin(GPIOD, B\_Channel\_Pin, GPIO\_PIN\_RESET);  HAL\_SPI\_Transmit(&hspi2, (uint8\_t\*)data, 16, 10);  HAL\_GPIO\_WritePin(GPIOD, SCLK\_Pin, GPIO\_PIN\_SET);  HAL\_GPIO\_WritePin(GPIOD, nOE\_Pin, GPIO\_PIN\_SET);  delay++;  if (a == 32)  { a=31;}  if (b == 16)  { b=15;}  if (a > 100)  { a=0;}  if (b > 100)  {b=0;}  if((AdcX > 3180) && (AdcY < 3235 && AdcY > 2700)){ // вправо  if(delay > 1000)  {  a = a+1;  delay = 0;  }  }  if((AdcX < 2700) && (AdcY < 3235 && AdcY > 2700)){ // влево  if(delay > 1000)  {  a = a-1;  delay = 0;  }  }  if((AdcX < 3180 && AdcX > 2700) && (AdcY < 2700)){ // вверх  if(delay > 1000)  {  b = b+1;  delay = 0;  }  }  if((AdcX < 3180 && AdcX > 2700) && (AdcY > 3235)){ // вниз  if(delay > 1000)  {  b = b-1;  delay = 0;  }  }  if((AdcX > 3180) && (AdcY < 2700)){ // вправо-вверх  if(delay > 1000)  {  a = a+1;  b = b+1;  delay = 0;  }  }  if((AdcX < 2700) && (AdcY < 2700)){ // влево-вверх  if(delay > 1000)  {  a = a-1;  b = b+1;  delay = 0;  }  }  if((AdcX > 3180) && (AdcY > 3235)){ // вправо-вниз  if(delay > 1000)  {  a = a+1;  b = b-1;  delay = 0;  }  }  if((AdcX < 2700) && (AdcY > 3235)){ // влево-вниз  if(delay > 1000)  {  a = a-1;  b = b-1;  delay = 0;  }  }  }  /\* USER CODE END WHILE \*/  /\* USER CODE BEGIN 3 \*/  }  /\* USER CODE END 3 \*/  } |

**STMStudio**

Выполним проверку работоспособности программы, приведя графики из STMStudio.

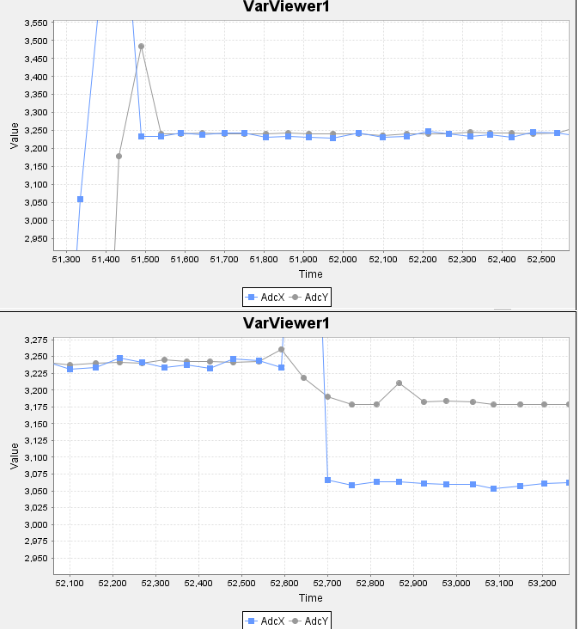


Рисунок 7 – Отслеживание аналогового сигнала с джойстика в различных положениях

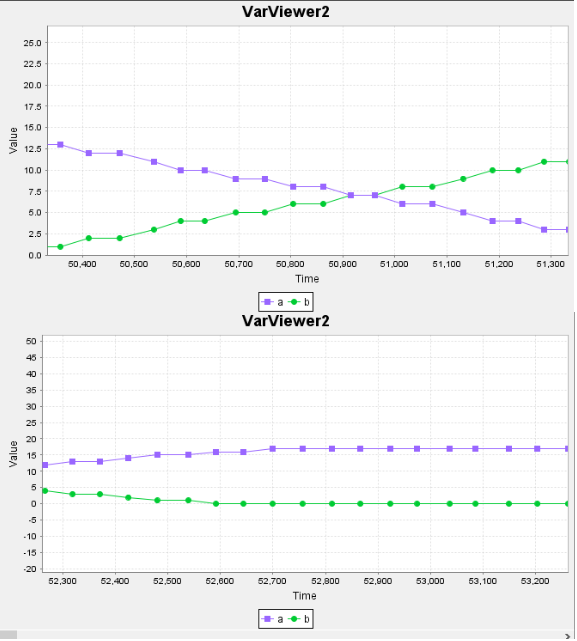


Рисунок 8 – Отслеживание координаты «мыши» при движение джойстика

**Результат работы**

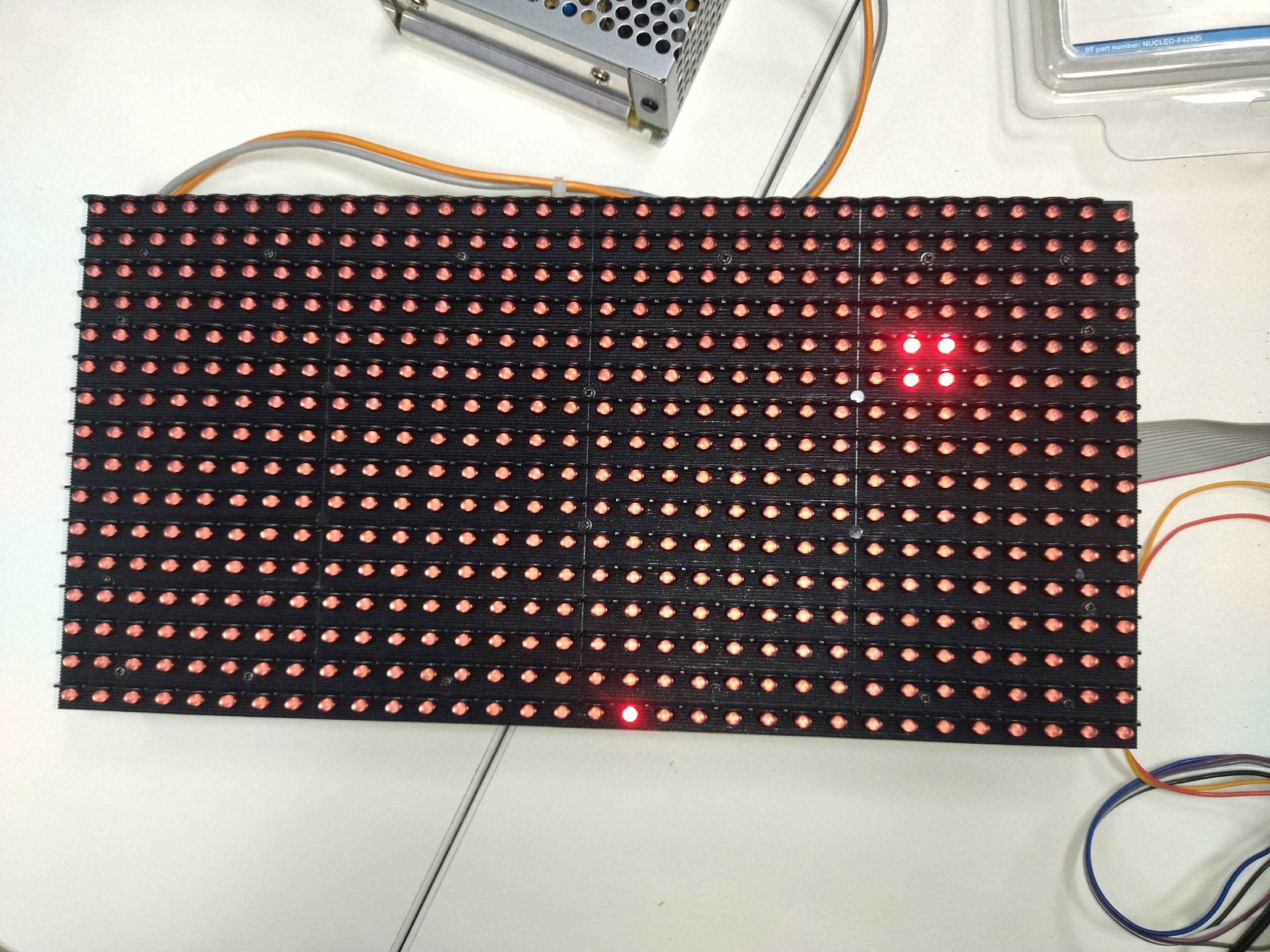


Рисунок 9 – Отрисованный объект и точка

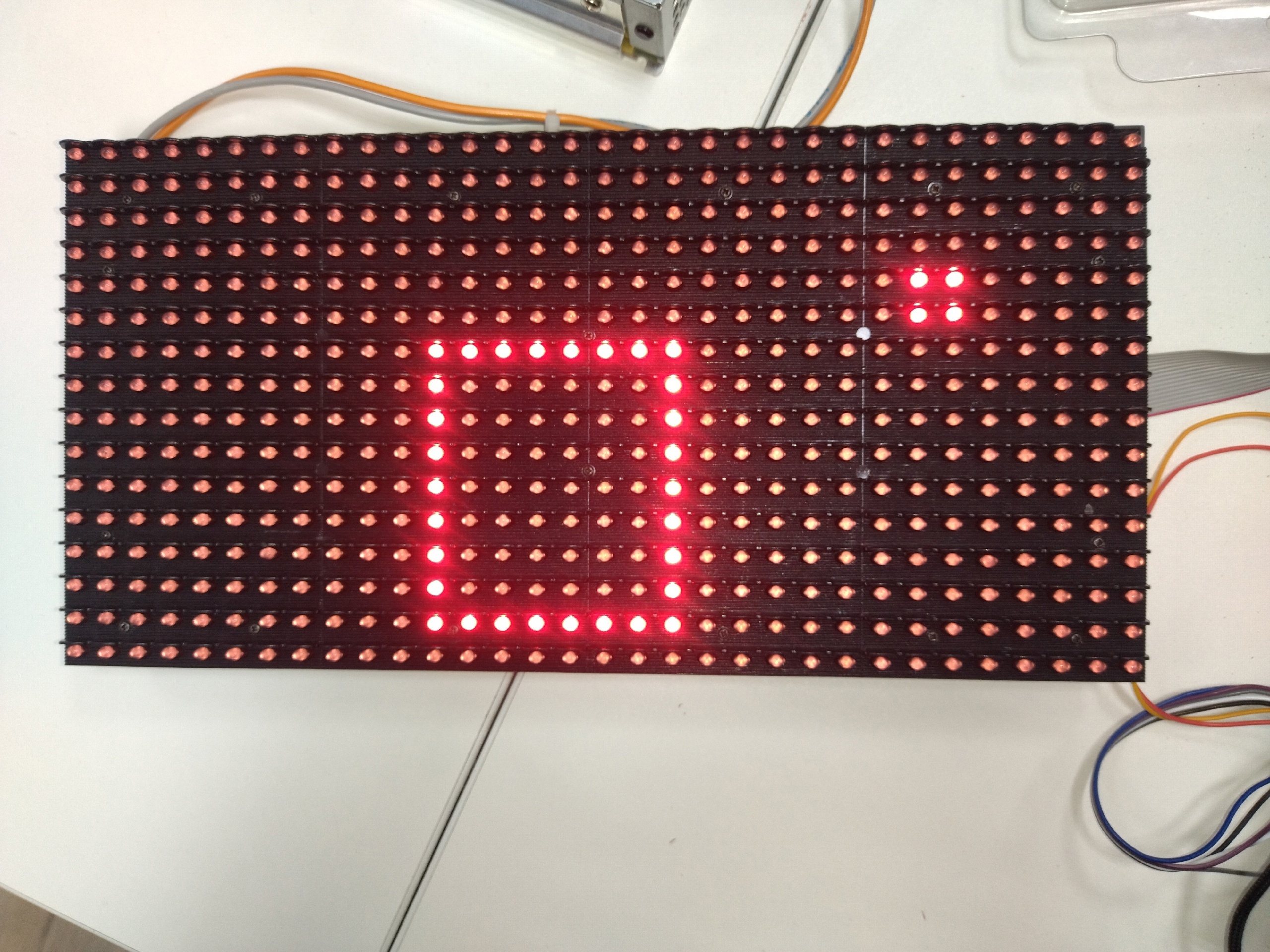


Рисунок 10 – Отрисованный объект и выделенное поле



Рисунок 11 – Процесс переноса объекта с визуальной индикацией

Выводы по работе

В рамках лабораторной работы была изучена схема светодиодной матрицы LEDP10, особенности протокола SPI. Также была реализована имитация рабочего стола с помощью светодиодной панели и джойстика.