Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
(ФГАОУ ВО «ЮФУ»)

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине «Введение в инженерную деятельность»

«Работа с аппаратными устройствами связи с объектом»

Выполнил
студент КТбо1-4 _____ А. А. Воронов

Принял
ассистент ИКТИБ _____ А. А. Зубкова

Таганрог 2024

ВВЕДЕНИЕ

Цели и задачи:

Целью работы является изучение способов связи с аппаратными

устройствами, получение информации из датчиков и управление драйверами.

Задачами работы являются:

1) Управление двигателем постоянного тока;

2) Настройка регулятора напряжения и считывание его параметров;

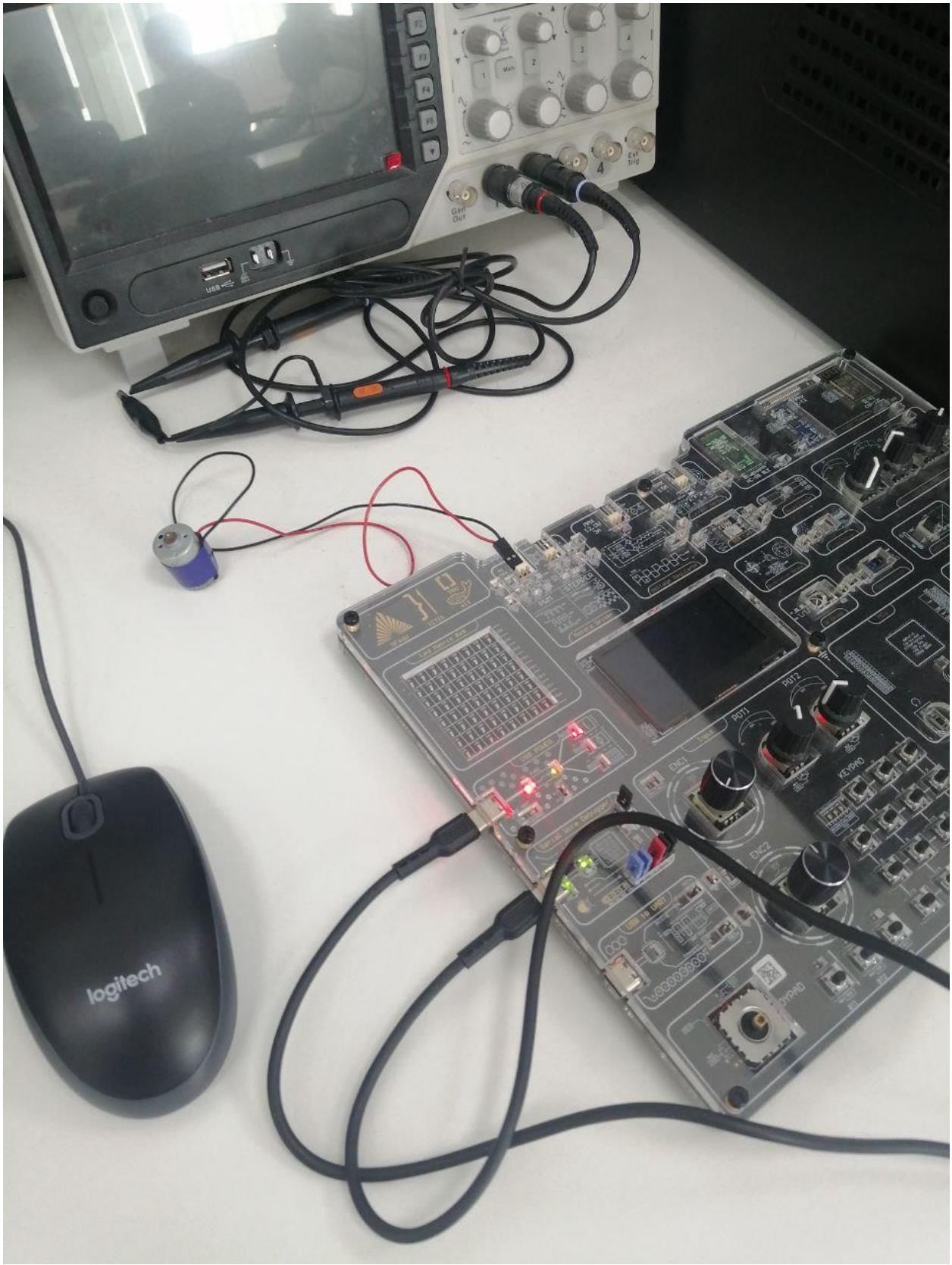
3) Получение показаний аналоговых датчиков;

4) Подключение и приём данных от цифровых сенсоров;

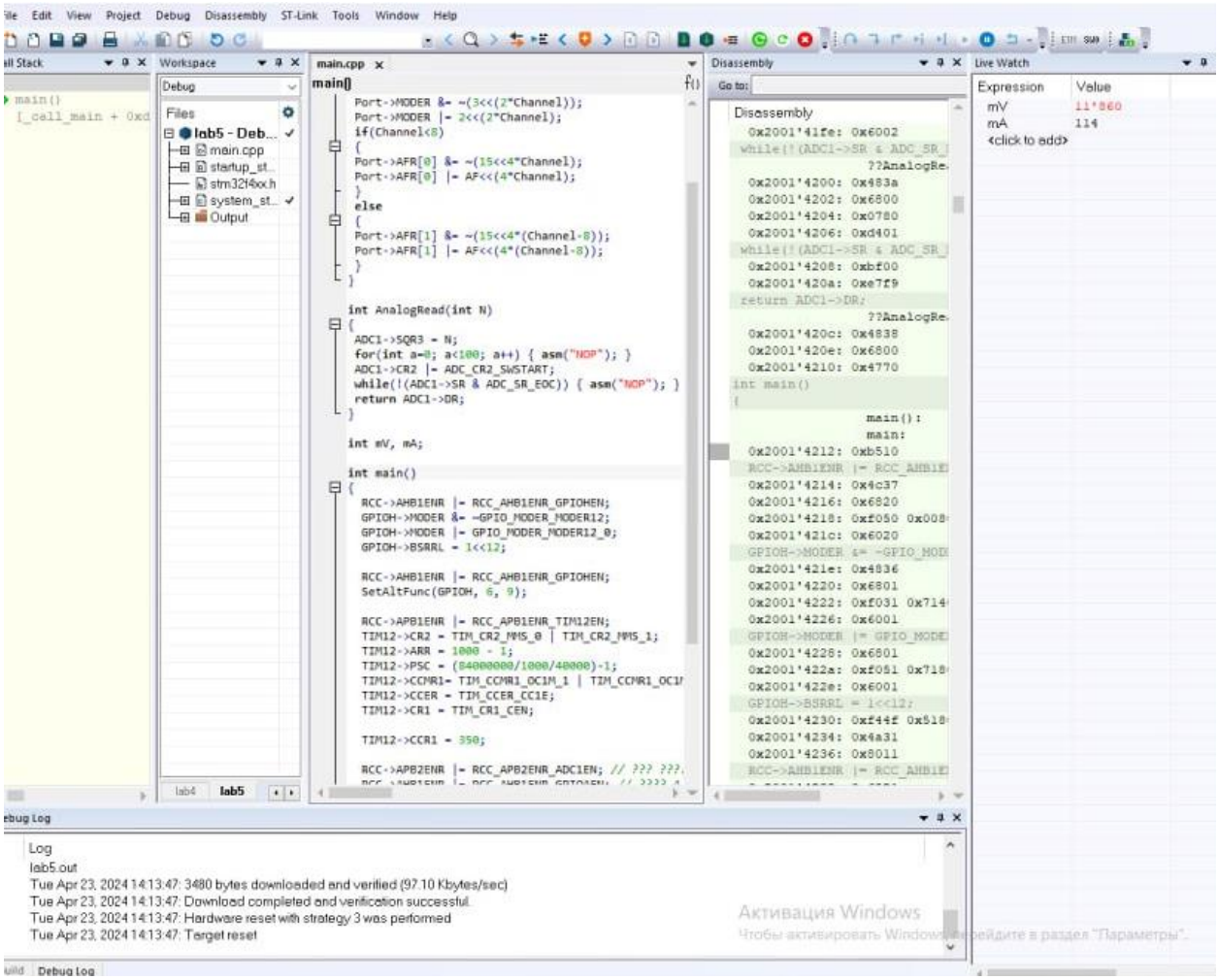
5) Отображение информации на OLED экране.

ОСНОВНАЯ ЧАСТЬ

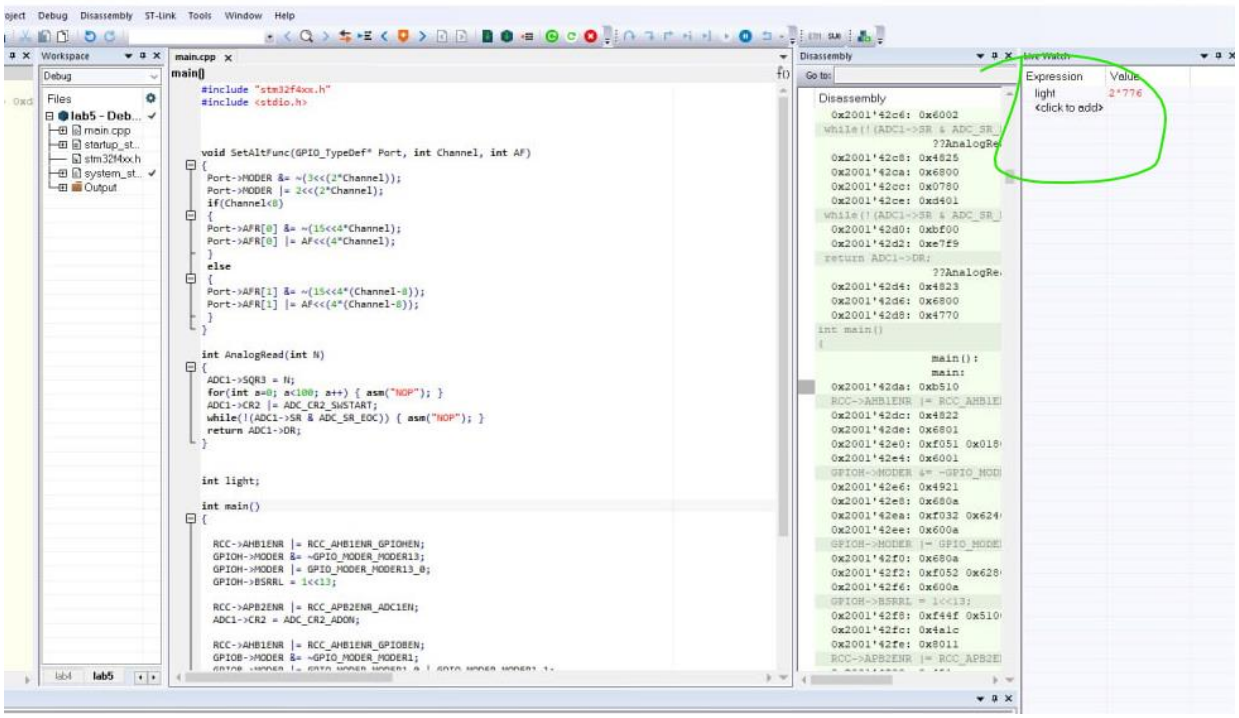
1. Запуск вибрации моторчика

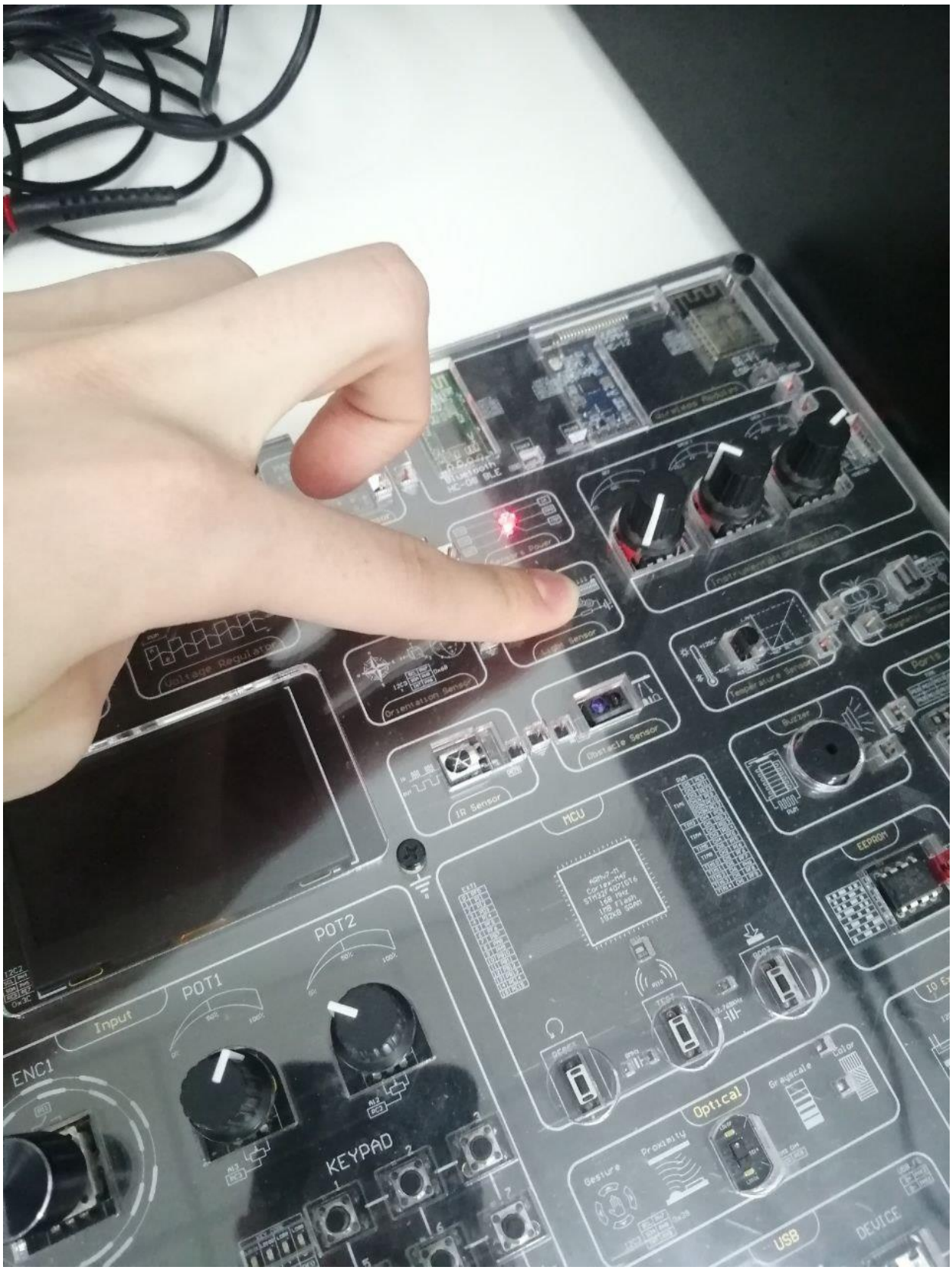


2. Получение значений напряжения и силы тока стабилизатора. Регуляция энкодером.

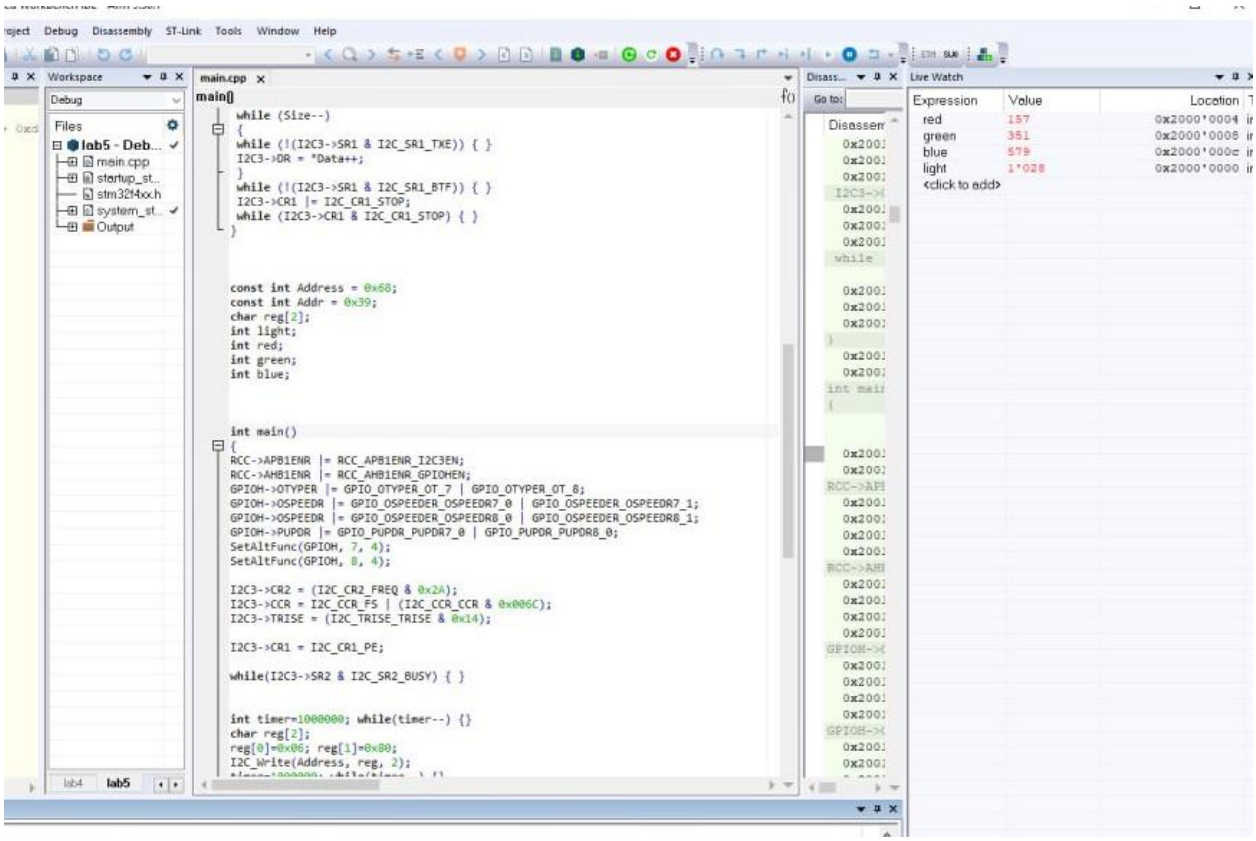


3. Считывание показаний датчика света.

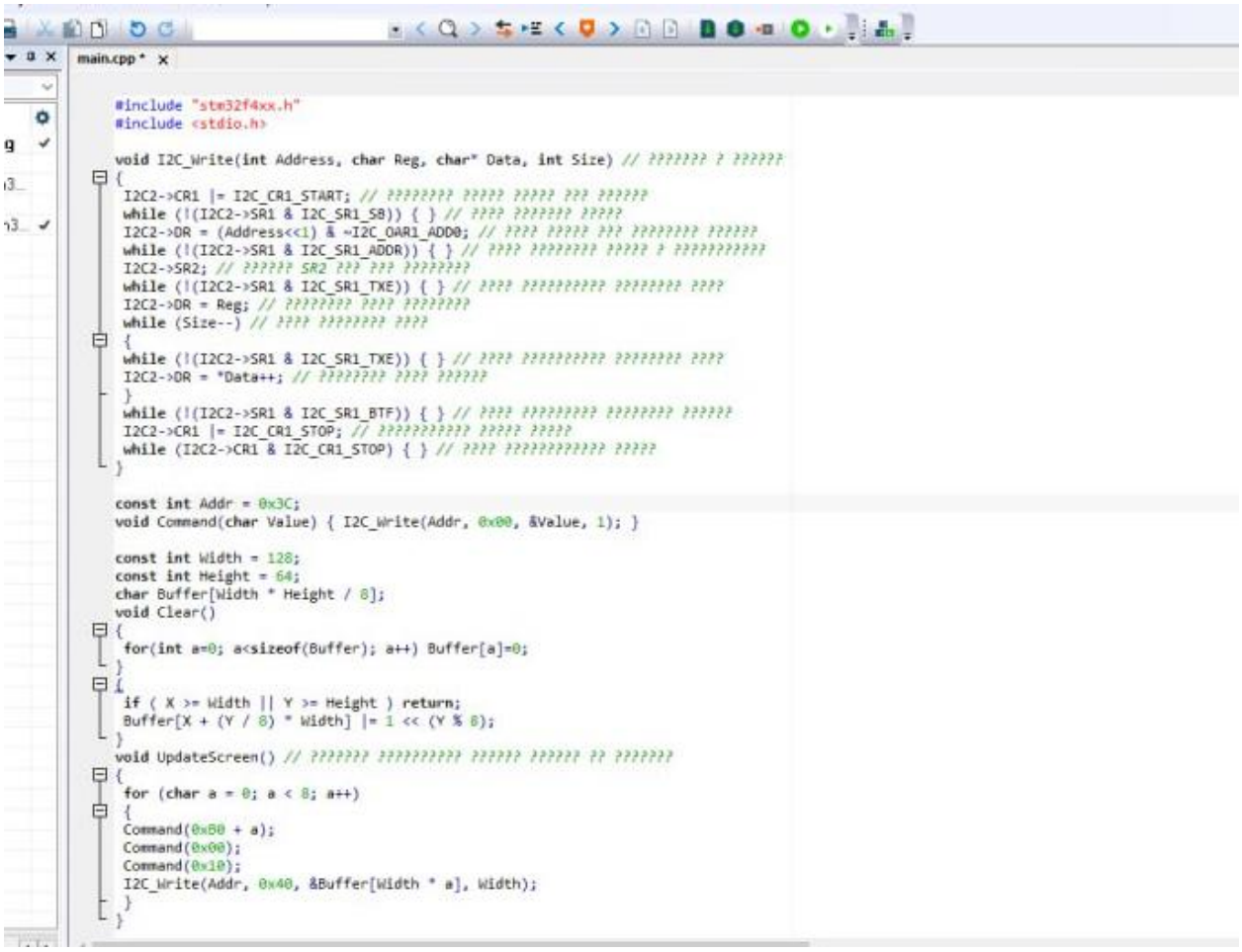

Debug
Disassembly
ST-Link
Tools
Window
Help
Workspace
Debug
Files
lab5 - Deb...
main.cpp
startup_st...
stm32f4xx.h
system_st...
Output
lab4
lab5
main.cpp
main()
#include "stm32f4xx.h"
#include <stdio.h>
void SetAltFunc(GPIO_TypeDef* Port, int Channel, int AF)
{
Port->MODER &= ~(3<<(2*Channel));
Port->MODER |= 2<<(2*Channel);
if(Channel<8)
{
Port->AFR[0] &= ~(15<<4*Channel);
Port->AFR[0] |= AF<<(4*Channel);
}
else
{
Port->AFR[1] &= ~(15<<4*(Channel-8));
Port->AFR[1] |= AF<<(4*(Channel-8));
}
}
int AnalogRead(int N)
{
ADC1->SQR3 = N;
for(int a=0; a<100; a++) { asm("NOP"); }
ADC1->CR2 |= ADC_CR2_SWSTART;
while(!(ADC1->SR & ADC_SR_EOC)) { asm("NOP"); }
return ADC1->DR;
}
int light;
int main()
{
RCC->AHB1ENR |= RCC_AHB1ENR_GPIOHEN;
GPIOH->MODER &= ~GPIO_MODER_MODER13;
GPIOH->MODER |= GPIO_MODER_MODER13_0;
GPIOH->BSRRL = 1<<13;
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN;
ADC1->CR2 = ADC_CR2_ADON;
RCC->AHB1ENR |= RCC_AHB1ENR_GPIOBEN;
GPIOB->MODER &= ~GPIO_MODER_MODER1;
Disassembly
Go to:
Disassembly
0x2001'42c6: 0x6002
while(!(ADC1->SR & ADC_SR_
??AnalogRe
0x2001'42c8: 0x4825
0x2001'42ca: 0x6800
0x2001'42cc: 0x0780
0x2001'42ce: 0xd401
while(!(ADC1->SR & ADC_SR_
0x2001'42d0: 0xbf00
0x2001'42d2: 0xe7f9
return ADC1->DR;
??AnalogRe
0x2001'42d4: 0x4823
0x2001'42d6: 0x6800
0x2001'42d8: 0x4770
int main()
{
main():
main:
0x2001'42da: 0xb510
RCC->AHB1ENR |= RCC_AHB1E
0x2001'42dc: 0x4822
0x2001'42de: 0x6801
0x2001'42e0: 0xf051 0x018
0x2001'42e4: 0x6001
GPIOH->MODER &= ~GPIO_MOD
0x2001'42e6: 0x4921
0x2001'42e8: 0x680a
0x2001'42ea: 0xf032 0x624
0x2001'42ee: 0x600a
GPIOH->MODER |= GPIO_MODE
0x2001'42f0: 0x680a
0x2001'42f2: 0xf052 0x628
0x2001'42f6: 0x600a
GPIOH->BSRRL = 1<<13;
0x2001'42f8: 0xf44f 0x510
0x2001'42fc: 0x4a1c
0x2001'42fe: 0x8011
RCC->APB2ENR |= RCC_APB2E
Live Watch
Expression
Value
light
2*776
<click to add>



4. Считывание показаний датчиков цвета.


Debug
Disassembly
ST-Link
Tools
Window
Help
Workspace
Debug
Files
lab5 - Deb...
main.cpp
startup_st...
stm32f4xx.h
system_st...
Output
lab4
lab5
main.cpp
main()
while (Size--)
{
while (!(I2C3->SR1 & I2C_SR1_TXE)) { }
I2C3->DR = *Data++;
}
while (!(I2C3->SR1 & I2C_SR1_BTF)) { }
I2C3->CR1 |= I2C_CR1_STOP;
while (I2C3->CR1 & I2C_CR1_STOP) { }
}
const int Address = 0x68;
const int Addr = 0x39;
char reg[2];
int light;
int red;
int green;
int blue;
int main()
{
RCC->APB1ENR |= RCC_APB1ENR_I2C3EN;
RCC->AHB1ENR |= RCC_AHB1ENR_GPIOHEN;
GPIOH->OTYPER |= GPIO_OTYPER_OT_7 | GPIO_OTYPER_OT_8;
GPIOH->OSPEEDR |= GPIO_OSPEEDER_OSPEEDR7_0 | GPIO_OSPEEDER_OSPEEDR7_1;
GPIOH->OSPEEDR |= GPIO_OSPEEDER_OSPEEDR8_0 | GPIO_OSPEEDER_OSPEEDR8_1;
GPIOH->PUPDR |= GPIO_PUPDR_PUPDR7_0 | GPIO_PUPDR_PUPDR8_0;
SetAltFunc(GPIOH, 7, 4);
SetAltFunc(GPIOH, 8, 4);
I2C3->CR2 = (I2C_CR2_FREQ & 0x2A);
I2C3->CCR = I2C_CCR_FS | (I2C_CCR_CCR & 0x006C);
I2C3->TRISE = (I2C_TRISE_TRISE & 0x14);
I2C3->CR1 = I2C_CR1_PE;
while(I2C3->SR2 & I2C_SR2_BUSY) { }
int timer=1000000; while(timer--) {}
char reg[2];
reg[0]=0x06; reg[1]=0x80;
I2C_Write(Address, reg, 2);
Disass...
Go to:
Live Watch
Expression
Value
Location
red
157
0x2000'0004
green
351
0x2000'0008
blue
579
0x2000'000c
light
1'028
0x2000'0000
<click to add>

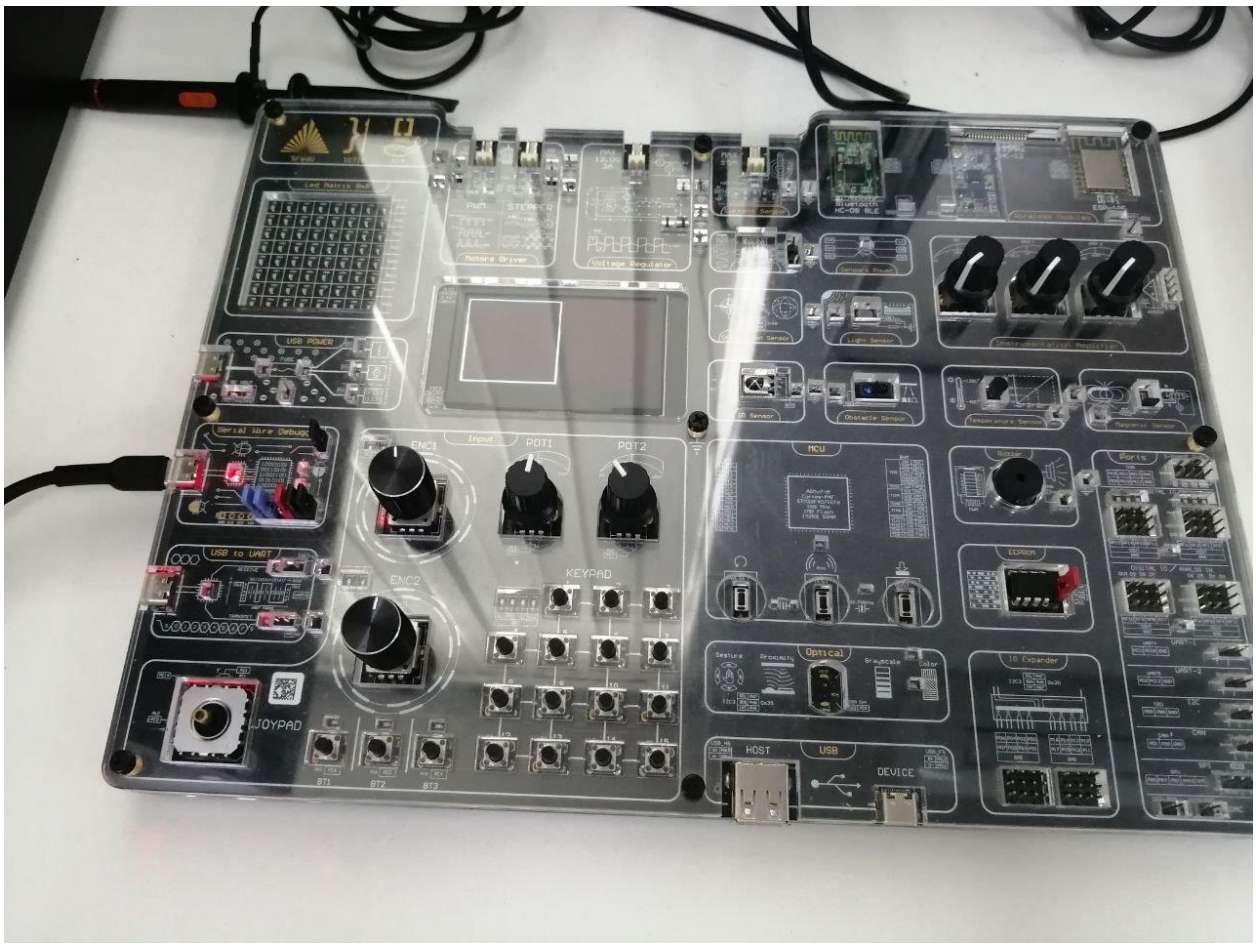
5. Вывод изображения на OLED дисплей.



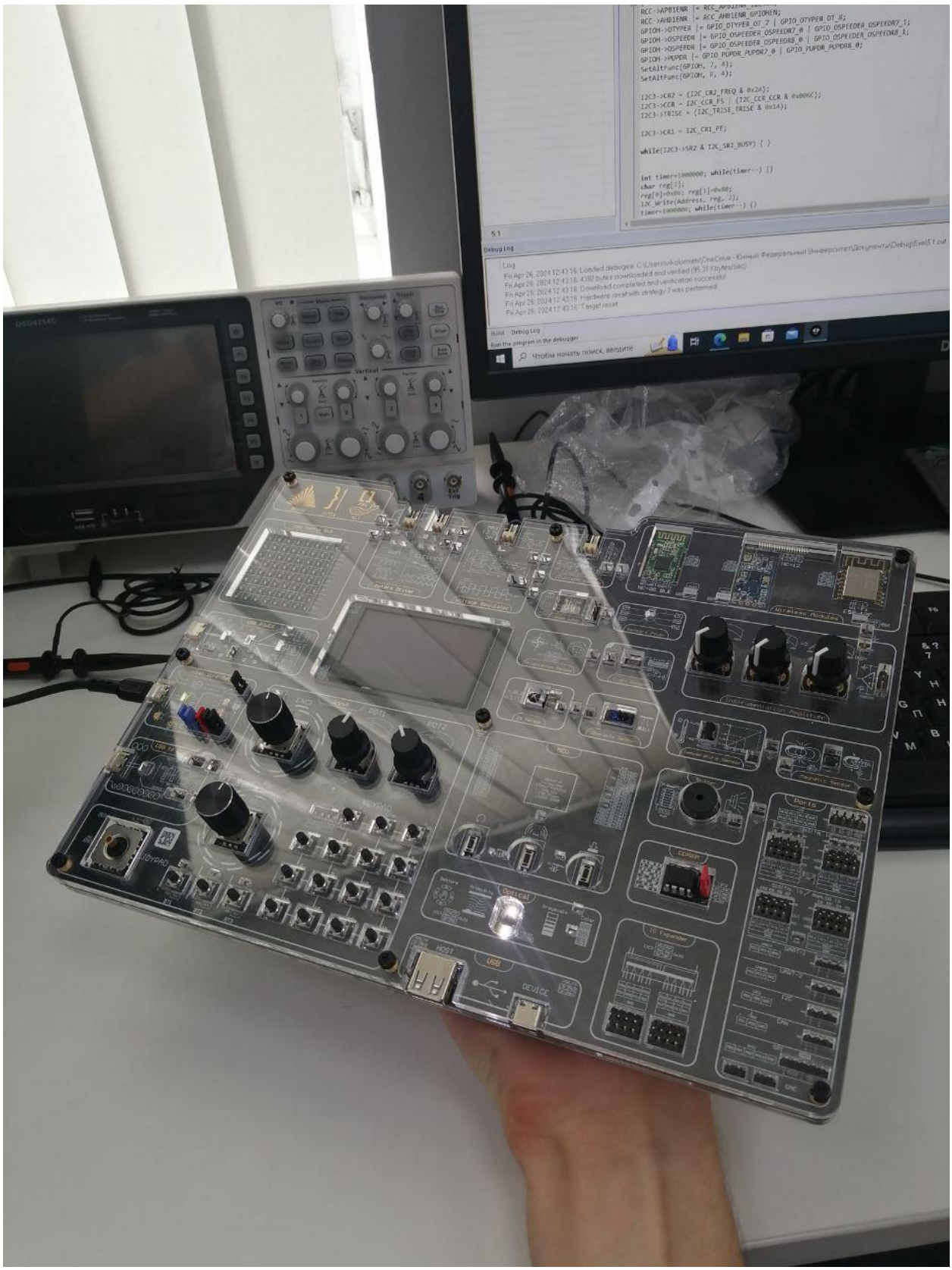
main.cpp *

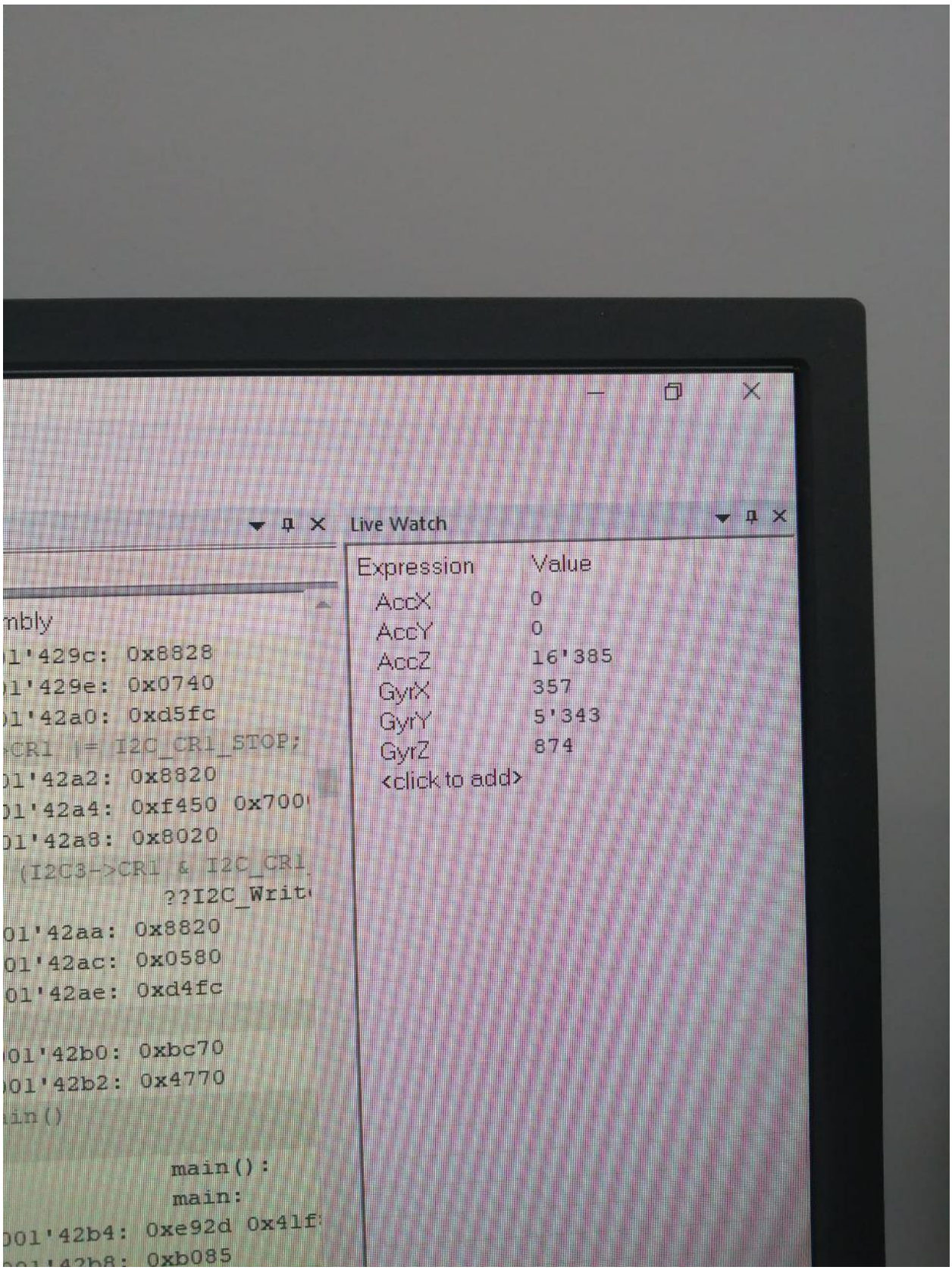
```
#include "stm32f4xx.h"
#include <stdio.h>

void I2C_Write(int Address, char Reg, char* Data, int Size) // ??????? ? ??????
{
 I2C2->CR1 |= I2C_CR1_START; // ???????? ????? ????? ??? ??????
 while (!(I2C2->SR1 & I2C_SR1_SB)) { } // ???? ??????? ?????
 I2C2->DR = (Address<<1) & ~I2C_OAR1_ADD0; // ???? ????? ??? ???????? ??????
 while (!(I2C2->SR1 & I2C_SR1_ADDR)) { } // ???? ???????? ????? ? ??????????
 I2C2->SR2; // ?????? SR2 ??? ??? ????????
 while (!(I2C2->SR1 & I2C_SR1_TXE)) { } // ???? ?????????? ???????? ????
 I2C2->DR = Reg; // ???????? ???? ????????
 while (Size--) // ???? ???????? ????
 {
 while (!(I2C2->SR1 & I2C_SR1_TXE)) { } // ???? ?????????? ???????? ????
 I2C2->DR = *Data++; // ???????? ???? ??????
 }
 while (!(I2C2->SR1 & I2C_SR1_BTF)) { } // ???? ????????? ???????? ??????
 I2C2->CR1 |= I2C_CR1_STOP; // ??????????? ????? ?????
 while (I2C2->CR1 & I2C_CR1_STOP) { } // ???? ???????????? ?????
}

const int Addr = 0x3C;
void Command(char Value) { I2C_Write(Addr, 0x00, &Value, 1); }

const int Width = 128;
const int Height = 64;
char Buffer[Width * Height / 8];
void Clear()
{
 for(int a=0; a<sizeof(Buffer); a++) Buffer[a]=0;
}
{
 if ( X >= Width || Y >= Height ) return;
 Buffer[X + (Y / 8) * Width] |= 1 << (Y % 8);
}
void UpdateScreen() // ??????? ?????????? ?????? ?????? ?? ???????
{
 for (char a = 0; a < 8; a++)
 {
 Command(0xB0 + a);
 Command(0x00);
 Command(0x10);
 I2C_Write(Addr, 0x40, &Buffer[Width * a], Width);
 }
}
```


PWM
STEPPER
Motors Driver
Voltage Regulator
USB POWER
MCU
Input
POT1
POT2
ENC1
ENC2
KEYPAD
USB to UART
JOYPAD
HOST
USB



6. Считывание показаний гироскопа


RCC->APB1ENR |= RCC_APB1ENR_GPIOHEN;
RCC->AHB1ENR |= GPIO_OTYPER_OT_7 | GPIO_OTYPER_OT_8;
GPIOH->OTYPER |= GPIO_OSPEEDER_OSPEEDR7_0 | GPIO_OSPEEDER_OSPEEDR7_1;
GPIOH->OSPEEDR |= GPIO_OSPEEDER_OSPEEDR8_0 | GPIO_OSPEEDER_OSPEEDR8_1;
GPIOH->OSPEEDR |= GPIO_PUPDR_PUPDR7_0 | GPIO_PUPDR_PUPDR8_0;
GPIOH->PUPDR |= GPIO_PUPDR_PUPDR7_0 | GPIO_PUPDR_PUPDR8_0;
SetAltFunc(GPIOH, 7, 4);
SetAltFunc(GPIOH, 8, 4);
I2C3->CR2 = (I2C_CR2_FREQ & 0x2A);
I2C3->CCR = I2C_CCR_FS | (I2C_CCR_CCR & 0x006C);
I2C3->TRISE = (I2C_TRISE_TRISE & 0x14);
I2C3->CR1 = I2C_CR1_PE;
while(I2C3->SR2 & I2C_SR2_BUSY) { }
int timer=1000000; while(timer--) {}
char reg[2];
reg[0]=0x00; reg[1]=0x00;
I2C_Write(Address, reg, 2);
timer=1000000; while(timer--) {}
Debug Log
Log
Fri Apr 26, 2024 12:43:16: Loaded debugger C:\Users\vkolomiets\OneDrive - Южный Федеральный Университет\Документы\Debug\Exe\5 f.out
Fri Apr 26, 2024 12:43:16: 4392 bytes downloaded and verified (95.31 Kbytes/sec)
Fri Apr 26, 2024 12:43:16: Download completed and verification successful
Fri Apr 26, 2024 12:43:16: Hardware reset with strategy 7 was performed
Fri Apr 26, 2024 12:43:16: Target reset
Build
Debug Log
Run the program in the debugger
Чтобы начать поиск, введите
Wireless Module
Optical
HOST
USB
DEVICE
IO Expander
CAN


Live Watch
Expression
Value
AccX
0
AccY
0
AccZ
16'385
GyrX
357
GyrY
5'343
GyrZ
874
<click to add>
mbly
1'429c: 0x8828
1'429e: 0x0740
1'42a0: 0xd5fc
CR1 |= I2C_CR1_STOP;
1'42a2: 0x8820
1'42a4: 0xf450 0x700
1'42a8: 0x8020
(I2C3->CR1 & I2C_CR1_
??I2C_Writ
1'42aa: 0x8820
1'42ac: 0x0580
1'42ae: 0xd4fc
1'42b0: 0xbc70
1'42b2: 0x4770
in()
main():
main:
1'42b4: 0xe92d 0x41f
1'42b8: 0xb085

ЗАКЛЮЧЕНИЕ

В ходе работы были выполнены все цели и задачи. Было изучено взаимодействие устройств аппаратной связи и работа драйверов. Были задействованы устройства: электродвигатель, OLED дисплей, акселерометр и гироскоп, энкодер и стабилизатор.