**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 5

**З дисципліни:** *“Основи інтернету речей”*

**На тему:** “ Дослідження роботи ультразвукового та інфрачервоного сенсорів відстані за допомогою середовища STM32CubeIDE та плати STM32F4 Discovery”

**Лектор:**

проф. каф. ПЗ

Фечан А. В.

**Виконав:**

ст. гр. ПЗ-42

Бурець В. В.

**Прийняв:**

проф. каф. ПЗ

Фечан А. В.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2021 р.

∑= \_\_\_\_\_ .

Львів – 2021

**Тема роботи:** Дослідити особливості роботи ультразвукового та інфрачервоного сенсорів відстані за допомогою середовища STM32CubeIDE та плати STM32F4 Discovery.

**Мета роботи:** Дослідити роботу сенсорів відстані пристроїв відображення інформації на прикладі символьного рідкокристалічного дисплею.

**TЕОРЕТИЧНІ ВІДОМОСТІ**

Ультразвуковий далекомір HC-SR04 - це вміщені на одну плату приймач і передавач ультразвукового сигналу. Випромінювач генерує сигнал, який, відбившись від перешкоди, потрапляє на приймач. Вимірявши час, за який сигнал проходить до об'єкта і назад, можна оцінити відстань. Крім самих приймача і передавача, на платі знаходиться ще і необхідна обв'язка, щоб зробити роботу з цим датчиком простою і зручною.

Датчик GP2Y0A21 - це інфрачервоний датчик відстані від Sharp, який може виявити об'єкт, розташований на відстані від 10 до 80 см перед ним. Цей діапазон відстаней встановлений через лінійність залежності аналогового сигналу. Датчик відстані GP2Y0A21 використовує інфрачервоне світло для обчислення відстані до об’єкта шляхом тріангуляції. Інфрачервоний світлодіод посилає світловий сигнал, невидимий неозброєним оком, який відбивається в присутності предмета. Стрічка з фоторезистом вловлює світло, що відбивається, і виводить кут відбиття, а отже, і відстань. Відповідно до документації датчика аналогове значення приходить в межах від 0 до 3,3 В.

**ЗАВДАННЯ**

1. Ознайомитись з технічними особливостями застосування ультразвукового та інфрачервоного сенсорів тиску   
2. Створити проект для одночасного визначення відстані обома сенсорами з виводом значень на екран дисплею.   
3. Провести вимірювання відстані для предмету чорного кольору, білого кольору та предмету з м’якою структурою (поролон).

**ХІД ВИКОНАННЯ**

Код мовою C, що реалізує індивідуальне завдання згідно варіанту в середовищі STM32CubeIDE:

Файл з ініціалізацією та логікою роботи з акустичним сенсором

**#include** "acoustic\_sensor.h"

**#define** TRIG\_TIME 10

**static** uint16\_t **acoustic\_sensor\_convert\_duration**(uint32\_t u32L\_duration);

**static** **void**\* **acoustic\_sensor\_idle\_handler**(acoustic\_sensor\_t\* objPL\_this);

**static** **void**\* **acoustic\_sensor\_trig\_start\_handler**(acoustic\_sensor\_t\* objPL\_this);

**static** **void**\* **acoustic\_sensor\_trig\_handler**(acoustic\_sensor\_t\* objPL\_this);

**static** **void**\* **acoustic\_sensor\_echo\_start\_handler**(acoustic\_sensor\_t\* objPL\_this);

**static** **void**\* **acoustic\_sensor\_echo\_handler**(acoustic\_sensor\_t\* objPL\_this);

**void** **acoustic\_sensor\_init**(acoustic\_sensor\_t\* objPL\_this, GPIO\_TypeDef \* objPL\_trig\_port,

uint16\_t u16L\_trig\_pin, GPIO\_TypeDef \* objPL\_echo\_port, uint16\_t u16L\_echo\_pin)

{

objPL\_this->objP\_echo\_port = objPL\_echo\_port;

objPL\_this->objP\_trig\_port = objPL\_trig\_port;

objPL\_this->u16\_echo\_pin = u16L\_echo\_pin;

objPL\_this->u16\_trig\_pin = u16L\_trig\_pin;

objPL\_this->b\_on\_off = false;

objPL\_this->u16\_dist = 0xFFFF;

objPL\_this->u32\_timer = 0;

objPL\_this->objP\_state\_handler = acoustic\_sensor\_idle\_handler;

}

**void** **acoustic\_sensor\_run**(acoustic\_sensor\_t\* objPL\_this)

{

**if** (objPL\_this->objP\_state\_handler != NULL)

{

objPL\_this->objP\_state\_handler = objPL\_this->objP\_state\_handler(objPL\_this);

}

}

**void** **acoustic\_sensor\_start**(acoustic\_sensor\_t\* objPL\_this)

{

objPL\_this->b\_on\_off = true;

}

**void** **acoustic\_sensor\_stop**(acoustic\_sensor\_t\* objPL\_this)

{

objPL\_this->b\_on\_off = false;

}

uint16\_t **acoustic\_sensor\_get\_distance**(acoustic\_sensor\_t\* objPL\_this)

{

**return** objPL\_this->u16\_dist;

}

**void**\* **acoustic\_sensor\_idle\_handler**(acoustic\_sensor\_t\* objPL\_this)

{

**if** (objPL\_this->b\_on\_off)

{

**return** acoustic\_sensor\_trig\_start\_handler;

}

**return** acoustic\_sensor\_idle\_handler;

}

**void**\* **acoustic\_sensor\_trig\_start\_handler**(acoustic\_sensor\_t\* objPL\_this)

{

objPL\_this->u32\_timer = delay\_get\_us\_tick();

HAL\_GPIO\_WritePin(objPL\_this->objP\_trig\_port, objPL\_this->u16\_trig\_pin, *GPIO\_PIN\_SET*);

**return** acoustic\_sensor\_trig\_handler;

}

**void**\* **acoustic\_sensor\_trig\_handler**(acoustic\_sensor\_t\* objPL\_this)

{

**if** (delay\_get\_us\_tick() - objPL\_this->u32\_timer > TRIG\_TIME)

{

HAL\_GPIO\_WritePin(objPL\_this->objP\_trig\_port, objPL\_this->u16\_trig\_pin, *GPIO\_PIN\_RESET*);

**return** acoustic\_sensor\_echo\_start\_handler;

}

**return** acoustic\_sensor\_trig\_handler;

}

**void**\* **acoustic\_sensor\_echo\_start\_handler**(acoustic\_sensor\_t\* objPL\_this)

{

uint8\_t u8\_pin\_state = HAL\_GPIO\_ReadPin(objPL\_this->objP\_echo\_port, objPL\_this->u16\_echo\_pin);

**if** (u8\_pin\_state)

{

objPL\_this->u32\_timer = delay\_get\_us\_tick();

**return** acoustic\_sensor\_echo\_handler;

}

**return** acoustic\_sensor\_echo\_start\_handler;

}

**void**\* **acoustic\_sensor\_echo\_handler**(acoustic\_sensor\_t\* objPL\_this)

{

uint8\_t u8\_pin\_state = HAL\_GPIO\_ReadPin(objPL\_this->objP\_echo\_port, objPL\_this->u16\_echo\_pin);

**if**(u8\_pin\_state)

{

**return** acoustic\_sensor\_echo\_handler;

}

uint32\_t u32L\_duration = delay\_get\_us\_tick() - objPL\_this->u32\_timer;

objPL\_this->u16\_dist = acoustic\_sensor\_convert\_duration(u32L\_duration);

acoustic\_sensor\_stop(objPL\_this);

**return** acoustic\_sensor\_idle\_handler;

}

uint16\_t **acoustic\_sensor\_convert\_duration**(uint32\_t u32L\_duration)

{

**return** (u32L\_duration \* 10) / 58;

}

Файл з ініціалізацією та логікою роботи з інфрачервоним сенсором

**#include** <ir\_sensor.h>

**#include** <delay.h>

**static** **void**\* **ir\_sensor\_start\_dalay\_handler**(ir\_sensor\_t\* objPL\_this);

**static** **void**\* **ir\_sensor\_idle\_handler**(ir\_sensor\_t\* objPL\_this);

**static** **void**\* **ir\_sensor\_work\_handler**(ir\_sensor\_t\* objPL\_this);

**static** **void**\* **ir\_sensor\_wait\_adc\_handler**(ir\_sensor\_t\* objPL\_this);

**void** **ir\_sensor\_init**(ir\_sensor\_t\* objPL\_this, ADC\_HandleTypeDef\* objPL\_adc)

{

objPL\_this->u16\_adc\_val = TOO\_FAR\_DIST;

objPL\_this->objP\_adc = objPL\_adc;

objPL\_this->u32\_timer = delay\_get\_us\_tick();

objPL\_this->b\_on\_off = false;

objPL\_this->objP\_state\_handler = ir\_sensor\_start\_dalay\_handler;

math\_function\_2d\_construct(&objPL\_this->obj\_func, objPS\_data, 11);

filter\_init(&objPL\_this->obj\_filter, FILTER\_COUNT);

}

**void** **ir\_sensor\_run**(ir\_sensor\_t\* objPL\_this)

{

**if** (objPL\_this->objP\_state\_handler != NULL)

{

objPL\_this->objP\_state\_handler = objPL\_this->objP\_state\_handler(objPL\_this);

}

}

uint16\_t **ir\_sensor\_get\_distance**(ir\_sensor\_t\* objPL\_this)

{

**if** (objPL\_this->u16\_adc\_val >= MAX\_ADC || objPL\_this->u16\_adc\_val <= MIN\_ADC)

{

**return** TOO\_FAR\_DIST;

}

**return** math\_function\_2d\_get\_y(&objPL\_this->obj\_func, objPL\_this->u16\_adc\_val);

}

**void**\* **ir\_sensor\_idle\_handler**(ir\_sensor\_t\* objPL\_this)

{

**return** objPL\_this->b\_on\_off ? ir\_sensor\_work\_handler : ir\_sensor\_idle\_handler;

}

**void**\* **ir\_sensor\_start\_dalay\_handler**(ir\_sensor\_t\* objPL\_this)

{

**if** (delay\_get\_us\_tick() - objPL\_this->u32\_timer > START\_DELAY)

{

**return** objPL\_this->b\_on\_off ? ir\_sensor\_work\_handler : ir\_sensor\_idle\_handler;

}

**return** ir\_sensor\_start\_dalay\_handler;

}

**void** **ir\_sensor\_start**(ir\_sensor\_t\* objPL\_this)

{

objPL\_this->b\_on\_off = true;

}

**void** **ir\_sensor\_stop**(ir\_sensor\_t\* objPL\_this)

{

objPL\_this->b\_on\_off = false;

}

**void**\* **ir\_sensor\_work\_handler**(ir\_sensor\_t\* objPL\_this)

{

**if** (!objPL\_this->b\_on\_off)

{

**return** ir\_sensor\_idle\_handler;

}

HAL\_ADC\_Start\_IT(objPL\_this->objP\_adc);

**return** ir\_sensor\_wait\_adc\_handler;

}

**void**\* **ir\_sensor\_wait\_adc\_handler**(ir\_sensor\_t\* objPL\_this)

{

**if** ((HAL\_ADC\_GetState(objPL\_this->objP\_adc) & HAL\_ADC\_STATE\_READY) == HAL\_ADC\_STATE\_READY)

{

uint16\_t u16L\_adc\_val = HAL\_ADC\_GetValue(objPL\_this->objP\_adc);

objPL\_this->u16\_adc\_val = filter\_calc(&objPL\_this->obj\_filter, u16L\_adc\_val);

ir\_sensor\_stop(objPL\_this);

**return** ir\_sensor\_idle\_handler;

}

**return** ir\_sensor\_wait\_adc\_handler;

}

Код у функції мейн  
/\* Initialize all configured peripherals \*/

MX\_GPIO\_Init();

MX\_I2C1\_Init();

MX\_I2S3\_Init();

MX\_SPI1\_Init();

MX\_USB\_HOST\_Init();

MX\_ADC2\_Init();

MX\_TIM6\_Init();

/\* USER CODE BEGIN 2 \*/

ir\_sensor\_t obj\_ir\_sensor;

//acoustic\_sensor\_t obj\_ac\_sensor;

lcd\_obj\_t obj\_lcd;

delay\_init(&htim6);

ir\_sensor\_init(&obj\_ir\_sensor, &hadc1);

ir\_sensor\_start(&obj\_ir\_sensor);

//acoustic\_sensor\_init(&obj\_ac\_sensor, GPIOC, GPIO\_PIN\_14, GPIOC, GPIO\_PIN\_15);

//acoustic\_sensor\_start(&obj\_ac\_sensor);

lcd\_init(&obj\_lcd, GPIOE, GPIO\_PIN\_12, GPIO\_PIN\_13, GPIO\_PIN\_14, GPIO\_PIN\_15, GPIO\_PIN\_11, GPIO\_PIN\_10, GPIO\_PIN\_7);

//lcd\_write\_string(&obj\_lcd, "Hello");

uint32\_t u32L\_seconds\_timer = delay\_get\_ms\_tick();

uint32\_t u32L\_10\_times\_per\_seconds\_timer = delay\_get\_ms\_tick();

uint32\_t u32L\_100\_times\_per\_seconds\_timer = delay\_get\_ms\_tick();

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

ir\_sensor\_run(&obj\_ir\_sensor);

//acoustic\_sensor\_run(&obj\_ac\_sensor);

lcd\_run(&obj\_lcd);

uint32\_t u32L\_crnt\_time = delay\_get\_ms\_tick();

**if** (u32L\_crnt\_time - u32L\_100\_times\_per\_seconds\_timer > 10)

{

ir\_sensor\_start(&obj\_ir\_sensor);

u32L\_100\_times\_per\_seconds\_timer = u32L\_crnt\_time;

}

**if** (u32L\_crnt\_time - u32L\_10\_times\_per\_seconds\_timer > 100)

{

ir\_sensor\_start(&obj\_ir\_sensor);

//acoustic\_sensor\_start(&obj\_ac\_sensor);

u32L\_10\_times\_per\_seconds\_timer = u32L\_crnt\_time;

}

**if** (u32L\_crnt\_time - u32L\_seconds\_timer > 1000)

{

u32L\_seconds\_timer = u32L\_crnt\_time;

u16\_ir\_dist = ir\_sensor\_get\_distance(&obj\_ir\_sensor);

//u16\_ac\_dist = acoustic\_sensor\_get\_distance(&obj\_ac\_sensor);

HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_15);

lcd\_write\_string(&obj\_lcd, "range" + u16\_ir\_dist);

}

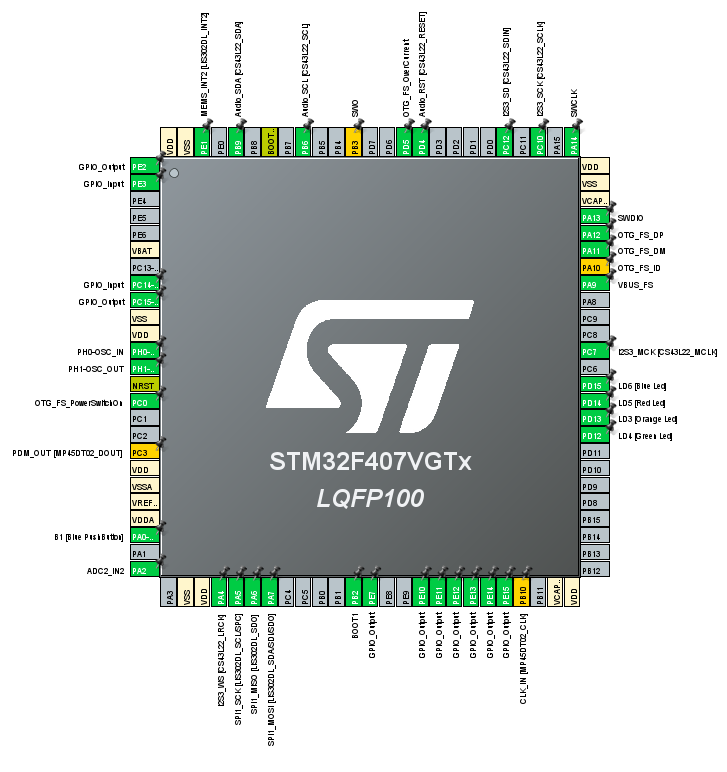
**РЕЗУЛЬТАТИ ВИКОНАННЯ**

Рис. 1. Налаштування пінів

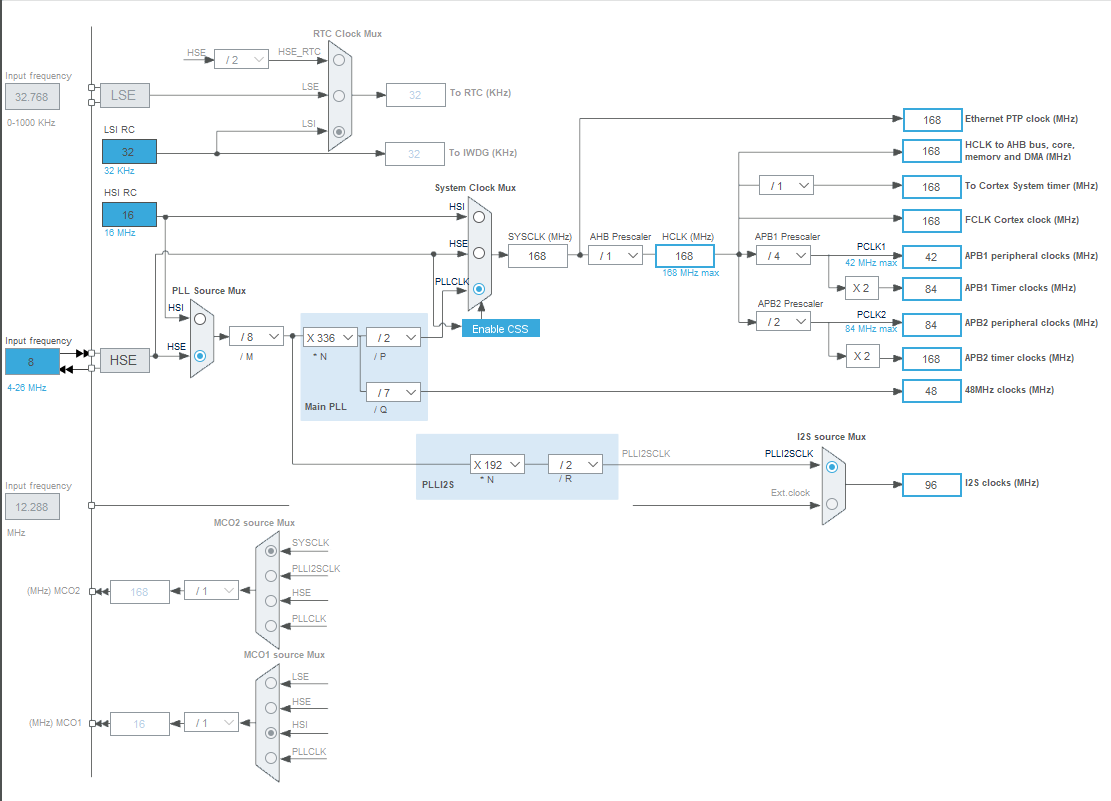
****

Рис. 2. Налаштування тактувань



Рис. 3. Показники інфрачервоного датчика відстані



Рис. 4. Показники акустичного датчика відстані

**ВИСНОВКИ**

На даній лабораторній роботі було набуто практичні навички у роботі з інфрачервоним та ультразвуковим датчиками відстані за допомогою середовища STM32CubeIDE та плати STM32F4 Discovery.

У результаті лабораторної роботи було досліджено, що  
предмети яскравих кольорів (червоного, жовтого, білого) здаються ближчими, ніж предмети темних кольорів (чорного, синього, сірого). Також результати було знайдено різницю у роботі датчиків. Вона пов’язана з робочим сигналом сенсора. Якщо це акустичний далекомір, то він буде видавати похибку при вимірюванні відстані до предмета з м'якою поверхнею, або на більшій відстані навіть перестане його бачити, оскільки відбитий акустичний сигнал буде частково поглинутий поверхню і відповідно послабиться, а відстані в цьому випадку визначається як час виявлення відбитого сигналу. Та сама історія з предметом з білою і чорною поверхнею і інфрачервоним датчиком. Світловий сигнал поглинатиметься темною поверхнею.

Також важливу роль має освітленість предмету, наприклад, якщо предмет знаходиться під прямими сонячними променями тоді відстань до нього теж визначатиметься з похибкою , оскільки відбите світло буде заглушати сигнал сенсора.