Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Програмування вбудованих систем*

Лабораторна робота № 9

Тема: «Робота з IR-пультами та приймачами»

Варіант : 16

Виконала: Петрів В. В.

Група ІПЗ-23

Дата: 19 грудня 2023р.

Викладач: Лазарович І.М.

Івано-Франківськ - 2023

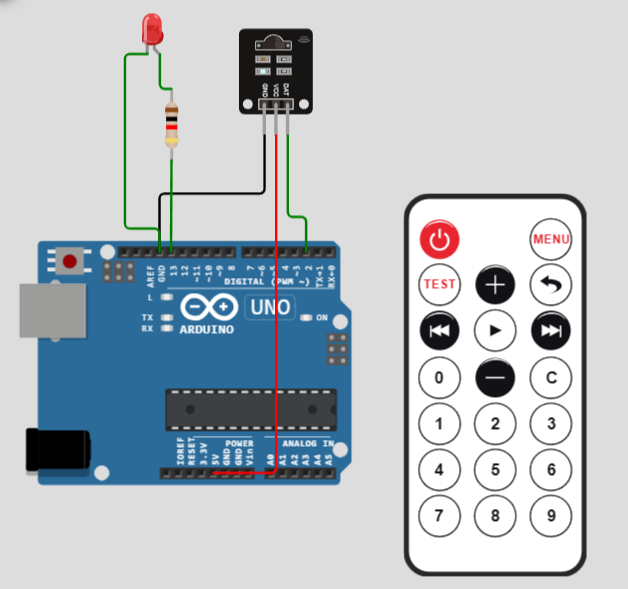
**Мета:** вивчити методику створення бібліотек функцій для платформи

Arduino.

**Завдання**

Використовуючи розроблені в лабораторній роботи №8 вихідні коди для роботи з інфрачервоним пультом дистанційного керування створити власну бібліотеку, назва якої побудована за шаблоном «IR\_Прізвище». Забезпечити: - підсвітку методів і класів; - вибір піну для сигналу з IR\_приймача; - змінну класу, яка стає true, якщо натиснута хоча б якась кнопка на пульті; - метод, який повертає десятковий код натиснутої кнопки; - метод, який перевіряє, чи була хоч раз натиснута кнопка, код якої він отримує як аргумент; - метод, який вмикає вбудований світлодіод після натиснення будь-якої кнопки на пульті; - метод, який після п’яти натиснень на OK починає миготіння вбудованого світлодіода з частотою 5Гц, використовуючи переривання. Після восьмого натискання на OK, миготіння (і світіння) треба припинити; - метод, який вмикає/вимикає виведення інформації про натиснуту кнопку в послідовний порт. Якщо це було перше натискання, то вивести тільки назву кнопки і її код, якщо ні, то виводити також кількість всіх натиснень.

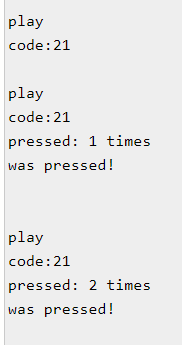
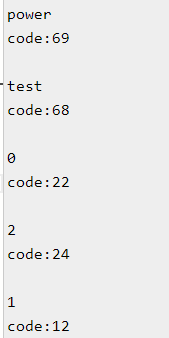
**Схема**

****

**Посилання на проект:**

<https://wokwi.com/projects/384665230621531137>

**Скрін виконання програми:**

****

**Програмний код**

#include "IR\_PETRIV.h"

IR\_PETRIV instance(2);

unsigned long timeOn = 200;

unsigned long changeTime = 0;

void setup() {

  pinMode (13, OUTPUT);

**Serial**.begin(9600);

  TCCR1A = 0;

  TCCR1B = 0;

  TCNT1  = 0;

  TIMSK1 = 1;

  instance.lightByAny();

}

ISR(TIMER1\_OVF\_vect) {

  instance.nec\_state = 0;

  TCCR1B = 0;

}

void loop() {

  if (instance.btnPressed()) {

    instance.decimal();

  }

}

**Програмний код IR\_PETRIV.h**

#ifndef IR\_PETRIV\_h

#define IR\_PETRIV\_h

#include "Arduino.h"

static void (\*\_\_timer1Hook)(void);

class IR\_PETRIV

{

  static IR\_PETRIV\* instance;

  static void remote\_read();

  public:

    IR\_PETRIV(int pin);

    bool btnPressed();

    int decimal();

    bool wasBtnPessed(byte dec);

    void lightByAny();

    void changeOutputMode(byte command);

    byte  nec\_state = 0;

  private:

    int \_pin;

    bool lightByAnyMod = false;

    int counter = 0;

    byte lastCommand = 0;

    byte  i, command, inv\_command;

    unsigned int address;

    unsigned long nec\_code;

    boolean nec\_ok;

};

#endif

**Програмний код IR\_PETRIV.cpp**

#include "Arduino.h"

#include "IR\_PETRIV.h"

byte codes[20] = {22, 12, 24, 94, 8, 28, 90, 66, 82, 74, 9, 7, 25, 64, 13, 67, 68, 71, 69, 21};

byte wasNotPressed[20] = {22, 12, 24, 94, 8, 28, 90, 66, 82, 74, 9, 7, 25, 64, 13, 67, 68, 71, 69, 21};

String values[20] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ">>", "<<", "-", "+", "c", "<-", "test", "menu", "power", "play" };

IR\_PETRIV\* IR\_PETRIV::instance = NULL;

IR\_PETRIV\* instance2 = NULL;

IR\_PETRIV::IR\_PETRIV(int pin)

{

**Serial**.begin(9600);

  instance = this;

  instance2= this;

  nec\_ok = 0;

  \_pin = pin;

  attachInterrupt(pin == 2 ? 0 : 1, remote\_read, CHANGE);

}

static void IR\_PETRIV::remote\_read() {

  unsigned int timer\_value;

  if (instance->nec\_state != 0) {

    timer\_value = TCNT1;

    TCNT1 = 0;

  }

  switch (instance->nec\_state) {

    case 0 :

      TCNT1  = 0;

      TCCR1B = 2;

      instance->nec\_state = 1;

      instance->i = 0;

      return;

    case 1 :

      if ((timer\_value > 19000) || (timer\_value < 17000)) {

        instance->nec\_state = 0;

        TCCR1B = 0;

      }

      else

        instance->nec\_state = 2;

      return;

    case 2 :

      if ((timer\_value > 10000) || (timer\_value < 8000)) {

        instance->nec\_state = 0;

        TCCR1B = 0;

      }

      else

        instance->nec\_state = 3;

      return;

    case 3 :

      if ((timer\_value > 1400) || (timer\_value < 800)) {

        TCCR1B = 0;

        instance->nec\_state = 0;

      }

      else

        instance->nec\_state = 4;

      return;

    case 4 :

      if ((timer\_value > 3600) || (timer\_value < 800)) {

        TCCR1B = 0;

        instance->nec\_state = 0;

        return;

      }

      if ( timer\_value > 2000)

        bitSet(instance->nec\_code, (31 - instance->i));

      else

        bitClear(instance->nec\_code, (31 - instance->i));

      instance->i++;

      if (instance->i > 31) {

        instance->nec\_ok = 1;

        detachInterrupt(0);

        return;

      }

      instance->nec\_state = 3;

  }

}

void IR\_PETRIV::lightByAny() {

  lightByAnyMod = true;

}

bool IR\_PETRIV::btnPressed() {

  return nec\_state;

}

void IR\_PETRIV::changeOutputMode(byte command) {

  if (command == lastCommand) {

    counter++;

  } else {

    counter = 0;

    lastCommand = command;

  }

}

bool IR\_PETRIV::wasBtnPessed(byte dec) {

  for (int i = 0; i < 20; i++){

    if (wasNotPressed[i] == dec ) {

      wasNotPressed[i] = -1;

      return false;

    }

  }

  return true;

}

IR\_PETRIV::decimal() {

  if (nec\_ok) {

    nec\_ok = 0;

    nec\_state = 0;

    TCCR1B = 0;

    command = nec\_code >> 8;

    changeOutputMode(command );

**Serial**.println();

    for (int i = 0; i < 20; i++){

      if (codes[i] == command ) {

**Serial**.println(values[i]);

      }

    }

**Serial**.print("code:");

**Serial**.println(command );

    if (counter) {

**Serial**.print("pressed: ");

**Serial**.print(counter);

**Serial**.print(" times\n");

    }

    if ( wasBtnPessed(command)){

**Serial**.println("was pressed! \n ");

    }

    if (lightByAnyMod){

      if(instance->counter > 8){

        digitalWrite(13, LOW);

      }else{

        digitalWrite(13, HIGH);

      }

    }

    attachInterrupt(\_pin == 2 ? 0 : 1, remote\_read, CHANGE);

  }

}

**Програмний код keywords.txt**

IR\_PETRIV KEYWORD1

changeOutputMode KEYWORD2

wasBtnPessed KEYWORD2

btnPressed KEYWORD2

lightByAny KEYWORD2

decimal KEYWORD2

**Висновок**

В даній лабораторній роботі ми навчились створювати власну бібліотеку для Arduino. Розробили бібліотеку для IR пульта дистанційного керування з підсвіткою методів, обранням піну для сигналу, перевіркою натискань, управлінням світлодіодом та іншими функціями.