



Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № R05

з курсу

”Мікропроцесорні та програмні засоби
автоматизації”

Робота з послідовним
інтерфейсом I2C за допомогою
WiringPi

Методичні вказівки до лабораторної роботи № R05 “Робота з послідовним інтерфейсом I2C за допомогою WiringPi” з курсу "Мікропроцесорні та програмні засоби автоматизації".
Медвідь В.Р., Пісьціо В.П., Тернопіль: ТНТУ, 20211 - 7 с.

Для студентів напрямку підготовки: 151 "Автоматизація та комп'ютерно-інтегровані технології"

Автори: Медвідь В.Р., Пісьціо В.П..

Тема роботи

Робота з послідовним інтерфейсом I2C за допомогою WiringPi

Мета роботи

Ознайомитись з можливостями та роботою із послідовними інтерфейсами Raspberry Pi за допомогою бібліотеки WiringPi.

Інтерфейс I2C

WiringPi включає бібліотеку, яка може полегшити використання інтерфейсу I2C у Raspberry Pi. Перш ніж використовувати інтерфейс I2C, можливо, вам доведеться скористатися утилітою `gpio` для завантаження драйверів I2C в ядро:

```
gpio load i2c
```

Якщо вам потрібна швидкість передачі даних, яка відрізняється від стандартних 100 Кбіт / с, тоді ви можете поставити це в командному рядку:

```
gpio load i2c 1000
```

встановить швидкість передачі даних в 1000Kbps - тобто. 1000000 bps. Для використання бібліотеки I2C вам потрібно:

```
#include <wiringPiI2C.h>
```

у вашій програмі. Програми повинні бути пов'язані з `-lwiringPi`, статично. Ви все ще можете використовувати стандартні системні команди для перевірки пристроїв I2C, і я рекомендую зробити це - наприклад, програма `i2cdetect`. Пам'ятайте лише, що на Rev 1 Raspberry pi - це пристрій 0, а на Rev. 2 - це пристрій 1. Наприклад

```
i2cdetect -y 0 # Rev 1
```

```
i2cdetect -y 1 # Rev 2
```

Зверніть увагу, що ви можете використовувати GPIO команду для запуску команди `i2cdetect` для вас з правильними параметрами для вашої ревізії плати:

```
gpio i2cdetect
```

`int wiringPiI2CSetup (int devId)`; ініціалізує шину I2C з вашим заданим ідентифікатором пристрою. Ідентифікатор - номер I2C пристрою, і ви можете використовувати програму `i2cdetect`, щоб дізнатися його. `wiringPiI2CSetup ()` розробить, яка редакція Raspberry Pi у вас є, і відкриє відповідний пристрій в `/ dev`. Повернене значення - це стандартний дескриптор файла Linux, або -1, якщо є помилка - у такому випадку ви можете звернутися до `errno`, як завжди.

Наприклад, популярним розширювачем GPIO MCP23017 зазвичай є Id пристрою 0x20, тому це число, яке ви переведете в `wiringPiI2CSetup ()`. Для всіх наступних функцій, якщо значення повернення від'ємне, сталася помилка, і вам слід звернутися до `errno`.

`int wiringPiI2CRead (int fd)`; Просте зчитування пристрою. Деякі пристрої представляють дані, коли ви їх читаєте, не здійснюючи жодних реєстраційних операцій.

`int wiringPiI2CWrite (int fd, int дані)`; простий запис пристрою. Деякі пристрої приймають дані таким чином, не потребуючи доступу до жодних внутрішніх регістрів.

```
int wiringPiI2CWriteReg8 (int fd, int reg, int дані);
```

`int wiringPiI2CWriteReg16 (int fd, int reg, int дані)`; функції записують 8 або 16-бітове значення даних у зазначений реєстр пристроїв.

```
int wiringPiI2CReadReg8 (int fd, int reg);
```

`int wiringPiI2CReadReg16 (int fd, int reg)`; функції читають 8 або 16-бітове значення із зазначеного реєстру пристроїв.

Робота із годинником реального часу

Для роботи із годинником реального часу PCF8563 необхідно використовувати плату ARPI600.

1 Переконайтесь, що джампер 17 встановлений у вірну позицію, так як показано на відповідному рисунку.

2. Відкрити термінал і у ньому запустити програму

```
i2cdetect -y 1
```

3 Переконайтесь, що PCF8563 з котрою відбувається зв'язок знаходиться за адресою 51. Що можна проконтролювати за появою коду 51 у виводі програми `i2cdetect`.

```

pi@raspberrypi i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: --- --- --- --- --- --- --- --- --- --- --- --- --- ---
10: --- --- --- --- --- --- --- --- --- --- --- --- --- ---
20: --- --- --- --- --- --- --- --- --- --- --- --- --- ---
30: --- --- --- --- --- --- --- --- --- --- UU --- --- ---
40: --- --- --- --- --- --- --- --- --- --- --- --- --- ---
50: --- 51 --- --- --- --- --- --- --- --- --- --- --- ---
60: --- --- --- --- --- --- --- --- --- --- --- --- --- ---
70: --- --- --- --- --- --- --- --- --- --- --- --- --- ---

```

Рис. 1. Вивід програми i2cdetect

Якщо все працює, за допомогою терміналу запустіть програму читання часу із годинника реального часу, що виводить дані на термінал

```

modprobe i2c-dev
echo pcf8563 0x51 > /sys/class/i2c-adapter/i2c-1/new_device
hwclock -r

```

Має з'явитись час з PCF8563, що може відрізнитись від часу на Raspbian Pi. Для встановлення часу у годиннику реального часу можна скористатись командою **hwclock -w (Write the time of the Raspbian into PCF8563)**

```
hwclock -r
```

Для встановлення часу у Raspberry Pi на основі годинника реального часу має бути виконана команда

```
hwclock -s
```

Якщо необхідно встановити модуль RTC у якості модуля, що буде доступний без додаткових дій у подальшому можна додати у файл

```

/boot/config.txt,
стрічку (без лапок)
"dtoverlay=i2c-rtc,param=pcf8563"

```

Приклад програми роботи із інтерфейсом I2C

Мікросхема годинника реального часу PCF8563 підключена до інтерфейса I2C із номером 1. Приклад простої програми, яка читає регістри годинника реального часу і виводить їх стан у термінал показана нижче.

```

#include <wiringPi.h>
#include <wiringPiI2C.h>
#include <stdio.h>
#define SEC 0x02
#define MIN 0x03
#define HOUR 0x04
#define DAY 0x05
#define WEEK 0x06
#define MONTH 0x07
#define YEAR 0x08
#define PCF8563_Address 0x51
#define reg 0x02
//seconds,minutes,hours,days,weekdays,months,yeas
char buf[]={0x00,0x47,0x11,0x19,0x05,0x06,0x15};
char *str[] ={"SUN","Mon","Tues","Wed","Thur","Fri","Sat"};
int fd,i;
void pcf8563SetTime()
{ for(i = 0;i < 7;i++)
  wiringPiI2CWriteReg8(fd,reg + i,buf[i]);
}
void pcf8563ReadTime()
{ for(i = 0;i < 7;i++)

```

```

        buf[i] = (char)wiringPiI2CReadReg8(fd,reg + i);
    }
int main()
{ if (wiringPiSetup() < 0) return 1;
  fd = wiringPiI2CSetup(PCF8563_Address);
  printf("PCF8564 Test Program ... \n");
  // pcf8563SetTime();
  while(1)
  { pcf8563ReadTime();
    buf[0] = buf[0]&0x7F; //sec
    buf[1] = buf[1]&0x7F; //min
    buf[2] = buf[2]&0x3F; //hour
    buf[3] = buf[3]&0x3F; //day
    buf[4] = buf[4]&0x07; //week
    buf[5] = buf[5]&0x1F; //mouth
    //year/month/day
    printf("20%02x/%02x/%02x  ",buf[6],buf[5],buf[3]);
    //hour:minute/second
    printf("%02x:%02x:%02x  ",buf[2],buf[1],buf[0]);
    //weekday
    printf("%s\n",str[(unsigned char)buf[4]]);
    delay(1000);
  }
}

```

Карта регістрів мікросхеми показана на наступному рисунку.

Address	Register name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	control_status_1	TEST1	0	STOP	0	TESTC	0	0	0
01h	control_status_2	0	0	0	TI_TP	AF	TF	AIE	TIE
02h	VL_seconds	VL	<seconds 00 to 59 coded in BCD>						
03h	minutes	x	<minutes 00 to 59 coded in BCD>						
04h	hours	x	x	<hours 00 to 23 coded in BCD>					
05h	days	x	x	<days 01 to 31 coded in BCD>					
06h	weekdays	x	x	x	x	x	<weekdays 0 to 6 in BCD>		
07h	century_months	C	x	x	<months 01 to 12 coded in BCD>				
08h	years	<years 00 to 99 coded in BCD>							
09h	minute_alarm	AE	<minute alarm 00 to 59 coded in BCD>						
0Ah	hour_alarm	AE	x	<hour alarm 00 to 23 coded in BCD>					
0Bh	day_alarm	AE	x	<day alarm 01 to 31 coded in BCD>					
0Ch	weekday_alarm	AE	x	x	x	x	<weekday alarm 0 to 6 in BCD>		
0Dh	CLKOUT_control	FE	x	x	x	x	x	FD1	FD0
0Eh	timer_control	TE	x	x	x	x	x	TD1	TD0
0Fh	timer	<timer countdown value>							

Рис. 2.Карта регістрів мікросхеми PCF8563.

Завдання

- 1 Завантажити та зібрати програму, наведену у прикладі.
- 2 Модифікувати програму таким чином, щоб зчитувались і виводились значення не байтів, котрі містить мікросхема, а дату та час.
- 3 Створити і запустити власну програму, що читає дані із регістрів мікросхеми і виводить їх у зручному вигляді.
- 4 Складіть звіт з лабораторної роботи.

Контрольні запитання

1. Які функції для роботи із інтерфейсом I2C ви знаєте?

2. Які дії виконує функція `wiringPiI2CReadReg8`?
3. Опишіть дані, що зберігає PCF8563.
4. Яке призначення програми `i2cdetect` ?

Література

1. Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. -СПб.: БХВ-Петербург, 2016. - 544 с.
2. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. -СПб.: БХВ-Петербург, 2016.-464 с.
3. Петин В.А. Микрокомпьютеры Raspberry Pi.Практическое руководство. СПб.: БХВ-Петербург, 2015. -240 с.

Зміст

Тема роботи	3
Мета роботи.....	3
Інтерфейс I2C	3
Робота із годинником реального часу.....	3
Приклад програми роботи із інтерфейсом I2C	4
Завдання	5
Контрольні запитання.....	5
Література	6