

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № R06

з курсу

”Мікропроцесорні та програмні засоби
автоматизації”

Інсталяція та використання
Code::Blocks та WxSmith для
побудови простої програми
керування

Методичні вказівки до лабораторної роботи № R05 "Інсталяція та використання Code::Blocks та WxSmith для побудови простої програми керування" з курсу "Мікропроцесорні та програмні засоби автоматизації". Медвідь В.Р., Пісьціо В.П., Тернопіль: ТНТУ, 2021 - 9 с.

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

Автори: Медвідь В.Р., Пісьціо В.П..

Тема роботи

Інсталяція та використання Code::Blocks та WxSmith для побудови простої програми керування

Мета роботи

Навчитись встановлювати, налаштовувати Code::Blocks і складати прості програми у ньому

Інсталяція та використання Code::Blocks та WxSmith

WxWidgets - це велика мультисистемна бібліотека для створення графічних інтерфейсів програм, що дозволяє створювати графічний інтерфейс програм і для Windows і для Linux. Для створення програм із даною бібліотекою зручно використовувати середовище розробки Code::Blocks IDE, що IDE може працювати в Windows і Linux.

Для запобігання проблем у процесі інсталяції перед встановленням останньої версії Code::Blocks IDE необхідно мати встановлений Midnight Commander, що зазвичай і так є у системі. Якщо його не має, то слід виконати команду

```
sudo apt-get install mc
```

або обрати програму через графічну оболонку інсталятора.

Проведемо встановлення CodeBlocks + WxWidgets на Raspbian.

1. Спочатку встановимо сам Code::Blocks IDE за допомогою консолі LXTerminal:

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install codeblocks
```

Ці самі дії можна зробити через графічну оболонку інсталятора обравши всі пакети, що пов'язані із Code::Blocks IDE.

2. Далі встановимо WxWidgets (бібліотека графічного інтерфейсу). Наступні бібліотеки повинні бути встановлені за допомогою консолі LXTerminal:

```
sudo apt install libwxgtk2.8-dev
```

Якщо бібліотека не знайдена можна спробувати написати

```
sudo apt install libwxgtk2.8.dev
```

Крім того буде потрібна бібліотека часу виконання

```
sudo apt install libwxgtk2.8.0
```

а також версія бібліотеки для відлагодження

```
sudo apt install libwxgtk2.8.dbg
```

Тепер все ще потрібен WxSmith

```
sudo apt install codeblocks-contrib
```

Також може виникнути потреба у інсталяції бібліотеки wiringPi, що потрібна для програмування інтерфейсу GPIO Raspberry Pi. Зазвичай вона інсталюється автоматично при розгортанні системи, але якщо її немає у консолі напишемо

```
sudo apt install wiringPi
```

Тепер можна запускати Code::Blocks. Запуск під root може бути потрібний для управління апаратурою через інтерфейс GPIO, якщо є така потреба запускатись тоді слід написати

```
sudo codeblocks
```

Якщо потрібні немає - можна відповідною іконкою у меню Пуск.

Створення нового проекту

Тепер можна створити новий проект за допомогою New - Project. І серед прототипів обрати проект wxWidgets

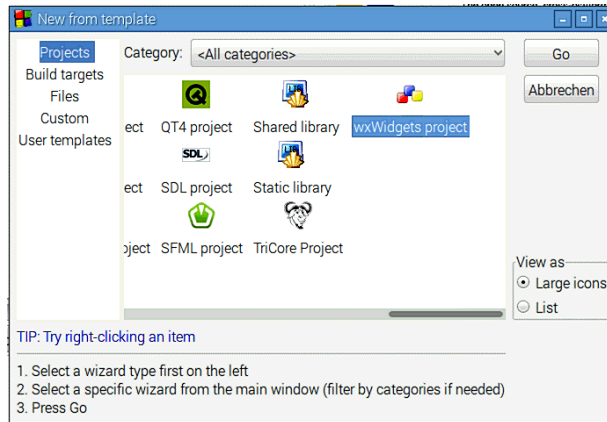


Рис. 1. Вибір типу проекту

Далі буде вікно привітання, (його можна відключити при наступних запусках) і після нього вікно вибору версії wxWidgets. Оберіть у ньому wxWidgets версії 2.8. (рис. 2).

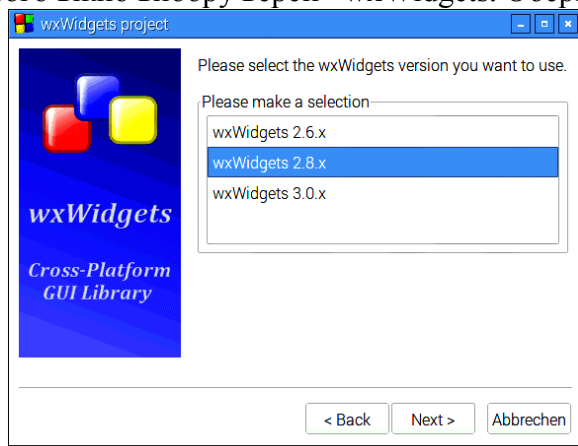


Рис. 2. Вибір версії бібліотеки

Далі слід обрати назву проекту та його розміщення (рис. 3) і інформацію про розробника рис. 4.

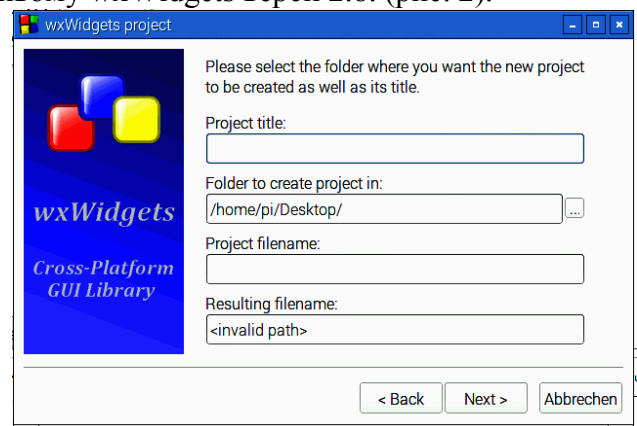


Рис. 3. Вибір розміщення

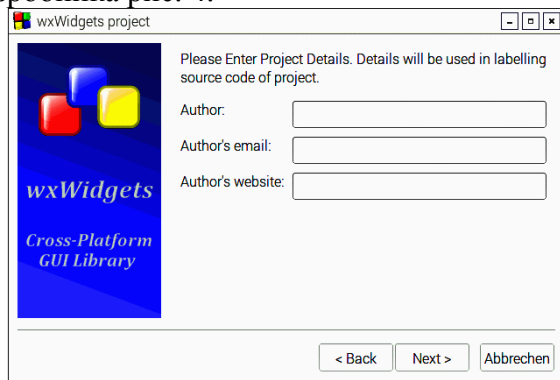


Рис. 4. Вибір інформації про розробника

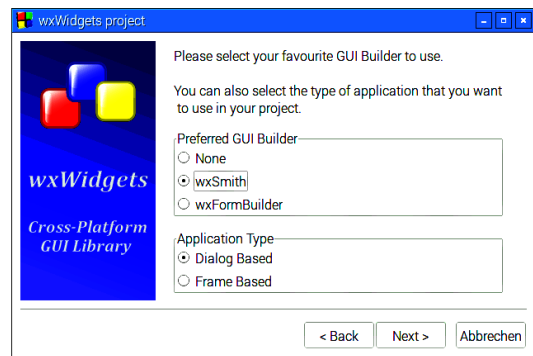


Рис. 5. Вибір конструктора і типу програми

Далі слід обрати wxSmith як конструктор графічного інтерфейсу і Dialog Based як тип програми (рис. 5). Для складних програм основаних на багатовіконному інтерфейсі слід використовувати тип програми Frame Based.

Далі необхідно становити конфігурацію компілятора і варіант використання бібліотек. Тобто використовувати бібліотеки як зовнішні модулі чи приєднати їх до програми. Для першої програми залишаємо налаштування за замовчуванням.

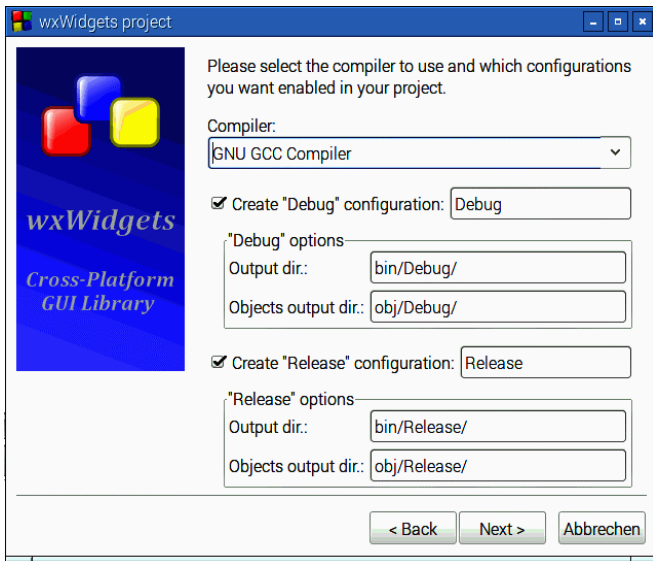


Рис. 6. Вибір налаштування компілятора

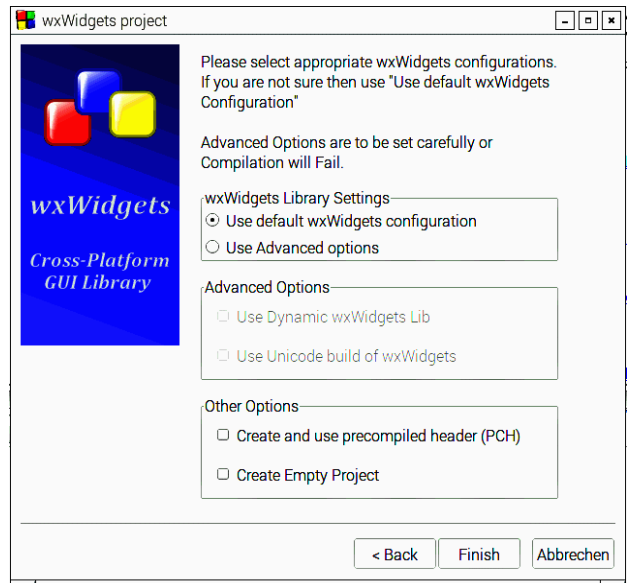


Рис. 7. Вибір варіанта використання бібліотек

Тепер слід натиснути кнопку Finish і дочекатись створення проекту. Коли проект створиться з'явиться демонстраційний код wxWidgets, і форма, що складається лише з двох кнопок About та Quit.

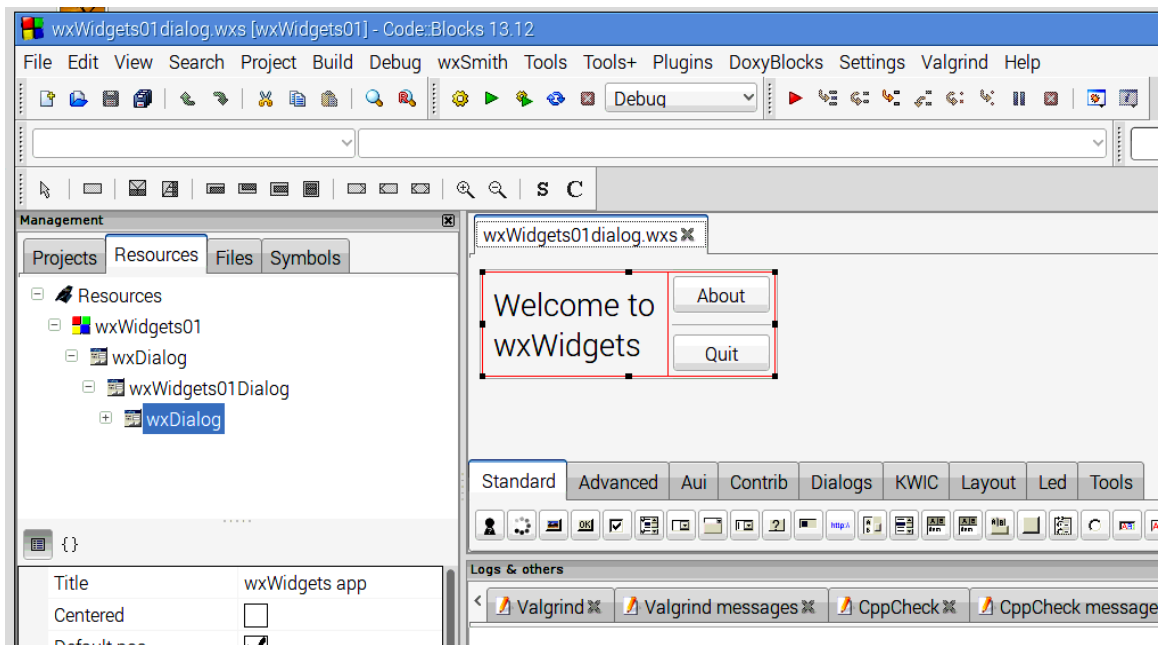


Рис. 8. Початкова програма

Після вибору "Build" -> "Build and Run" (це займе деякий час) має з'явитися невелике вікно - повністю скомпільована та запущена програма.

Оскільки будемо використовувати бібліотеку wiringPi, доведеться редагувати налаштування компілятора. Для цього обираємо Settings->Compiler... Відкриється вікно, фрагмент котрого зображений на наступному рисунку ()

У розділі Інші параметри лінкера вводимо -lwiringPi, що вказує компілятору підключати бібліотеку wiringPi

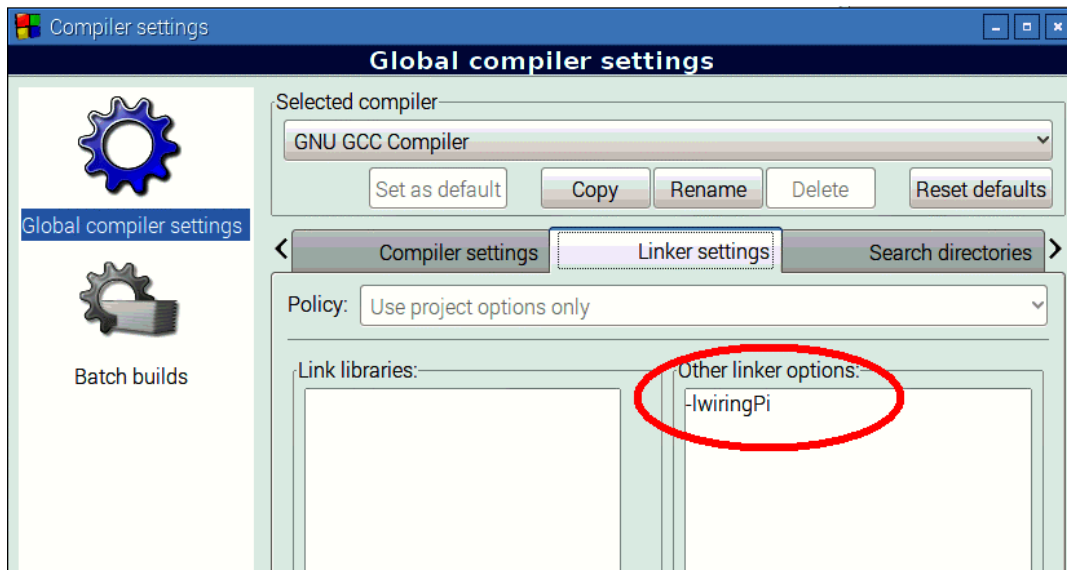


Рис. 9. Налаштування лінкера

Тепер все готово для створення проекту для мигання світлодіодом, що підключений до інтерфейса GPIO.

Написання найпростішої програми керування GPIO з використанням C та WxWidgets

Нехай ми хочемо керувати світлодіодом на виводі GPIO17 (wiringPi -> вивід 0, фізичний вивід 11) за допомогою графічного інтерфейсу (GUI).

Спочатку змінимо текстове поле з Welcome... на LED Control. Для цього обираємо wxStatic Text StaticText1. У полі властивостей зліва у низу екрану знаходимо поле Label і там змінюємо текст текстового поля на LED Control (зрозуміло, що може бути записаний будь-якій інший текст).

Аналогічним чином змінюємо текстові поля (поле Label) кнопок на LED ON та LED OFF. Зверніть увагу, що поля Var name та інші залишаємо без змін. Призначення полів можна знайти у довідникових матеріалах на wxwidgets.

Тепер із кнопками слід пов'язати дії, що мають бути виконані при їх натисканні. Так само як і у інших системах швидкої розробки програм, дії що виконуються пов'язуються із повідомленнями, що направляються системою кнопок. Такі повідомлення називають Events.

Отже обираємо Button1 і в вікні властивостей обираємо вкладку Events (рис. 10). Обираємо Add new handler. Пропонується назвати новий обробник подій OnButton1Click. Підтверджуємо за допомогою ОК (рис. 11):

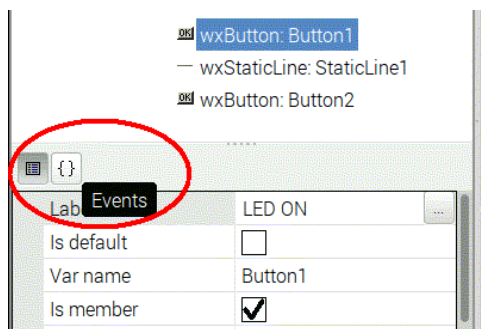


Рис. 10. Події об'єкта

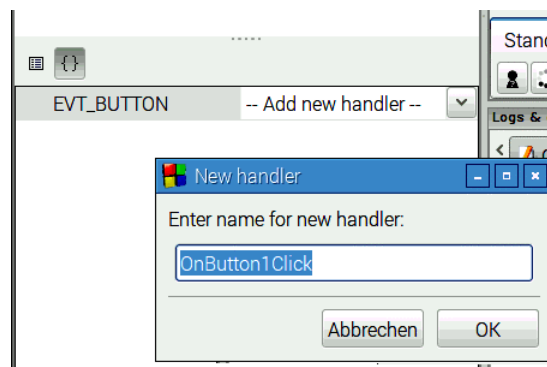


Рис. 11. Додавання нового обробника

У результаті CodeBlocks відкриває файл із назвою LED_BLINK_02Main.cpp, у котрий у самому низу коду тепер можна знайти обробник подій Button1 (світлодіод увімкнено):

```
void LED_BLINK_02Dialog::OnButton1Click(wxCommandEvent& event)
{
}
```

Добавляємо код для включення нашого світлодіода: digitalWrite (0, HIGH):

```
void LED_BLINK_02Dialog::OnButton1Click(wxCommandEvent& event)
```

```
{ digitalWrite(0, HIGH);
}
```

Код означає, що контакт 0 (wiringPi) при натисканні кнопки встановлюється у високий рівень (3,3 В).

Тепер аналогічно створюється обробник OnButton2Click для Button2 (OFF), а потім пишеться о код програми:

```
void LED_BLINK_02Dialog::OnButton2Click(wxCommandEvent& event)
{ digitalWrite(0, LOW);
}
```

Щоб програма почала працювати необхідно включити у програму бібліотеку wiringPi. Для цього переходимо до верхньої частини вкладки з програмним кодом і вставляємо після інших директив #include стрічку

```
#include <wiringPi.h>
```

Також запишемо код ініціалізації інтерфейсу GPIO, що будемо проводити при створенні форми, як показано на наступному рисунку (рис. 12). Доданий код позначений на рисунку рамкою.

```

57     END_EVENT_TABLE()
58
59     LED_BLINK_02Dialog::LED_BLINK_02Dialog(wxWindow
60     {
61         //(*Initialize(LED_BLINK_02Dialog)
62         Create(parent, id, _("wxWidgets app"), wxD
63         BoxSizer1 = new wxBoxSizer(wxHORIZONTAL);
64         StaticText1 = new wxStaticText(this, ID_ST
65         wxFont StaticText1Font = wxSystemSettings:
66         if ( !StaticText1Font.Ok() ) StaticText1Fo
67         StaticText1Font.SetPointSize(20);
68         StaticText1->SetFont(StaticText1Font);
69         BoxSizer1->Add(StaticText1, 1, wxALL|wxALI
70         BoxSizer2 = new wxBoxSizer(wxVERTICAL);
71         Button1 = new wxButton(this, ID_BUTTON1, _
72         BoxSizer2->Add(Button1, 1, wxALL|wxALIGN_C
73         StaticLine1 = new wxStaticLine(this, ID_ST
74         BoxSizer2->Add(StaticLine1, 0, wxALL|wxEXP
75         Button2 = new wxButton(this, ID_BUTTON2, _
76         BoxSizer1->Add(Button2, 1, wxALL|wxALIGN_C
77         BoxSizer1->Add(BoxSizer2, 0, wxALIGN_CENTE
78         SetSizer(BoxSizer1);
79         BoxSizer1->Fit(this);
80         BoxSizer1->SetSizeHints(this);
81
82         Connect(ID_BUTTON1,wxEVT_COMMAND_BUTTON_CL
83         Connect(ID_BUTTON2,wxEVT_COMMAND_BUTTON_CL
84         //*)
85
86         wiringPiSetup();
87         pinMode(0, OUTPUT);
88
89     }
90

```

Рис. 12. Код ініціалізації

Тепер обираємо Build - Build and Run і перевіряємо, чи все працює без помилок. Якщо код був скомпільований без помилок, на екрані з'являється проста програма із графічним інтерфейсом, з котрої можна керувати світлодіодом.

Тепер спробуємо, чи працює ця програма LED-BLINK незалежно від IDE Code::Blocks. Виходимо з Code::Blocks і запускаємо консоль LXTerminal.

```

pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ cd LED_BLINK_02
pi@raspberrypi:~/Desktop/LED_BLINK_02 $ cd bin
pi@raspberrypi:~/Desktop/LED_BLINK_02/bin $ cd Debug
pi@raspberrypi:~/Desktop/LED_BLINK_02/bin/Debug $ sudo ./LED_BLINK_02

```

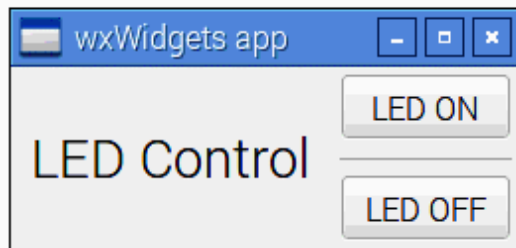


Рис. 13. Програма

Якщо все добре програма для керування знову з'явиться на робочому столі.

Завдання

1. Користуючись викладеними вище положеннями встановити і запустити програму Code::Blocks та встановити і налаштувати бібліотеку wxwidgets.
2. Набрати програму, що описана у методичних вказівках, відкомпілювати і запустити її.
3. Модифікувати програму, щоб вона відображала у полі StaticText1 стан світлодіода, котрим здійснюється керування.
- 4*. Ввести можливість вибору типу контакту, до якого підключений світлодіод. (Наприклад, випадаючим меню, або параметром командної строки чи іншим чином). Це завдання потребує ознайомлення із основними компонентами графічної системи і є творчим завдання більш високої складності.
5. Зробити висновки по роботі.

Література

1. WxWiki [Сайт] https://wiki.wxwidgets.org/Main_Page
2. Иго Т. Arduino, датчики и сети для связи устройств: Пер. с англ. -СПб.: БХВ-Петербург, 2016. - 544 с.
3. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. -СПб.: БХВ-Петербург, 2016.-464 с.
4. Петин В.А. Микрокомпьютеры Raspberry Pi.Практическое руководство. СПб.: БХВ-Петербург, 2015. -240 с.

Зміст

Тема роботи	3
Інсталяція та використання Code::Blocks та WxSmith	3
Створення нового проекту	3
Написання найпростішої програми керування GPIO з використанням C та WxWidgets ...	6
Завдання	8
Література	8
Зміст	9