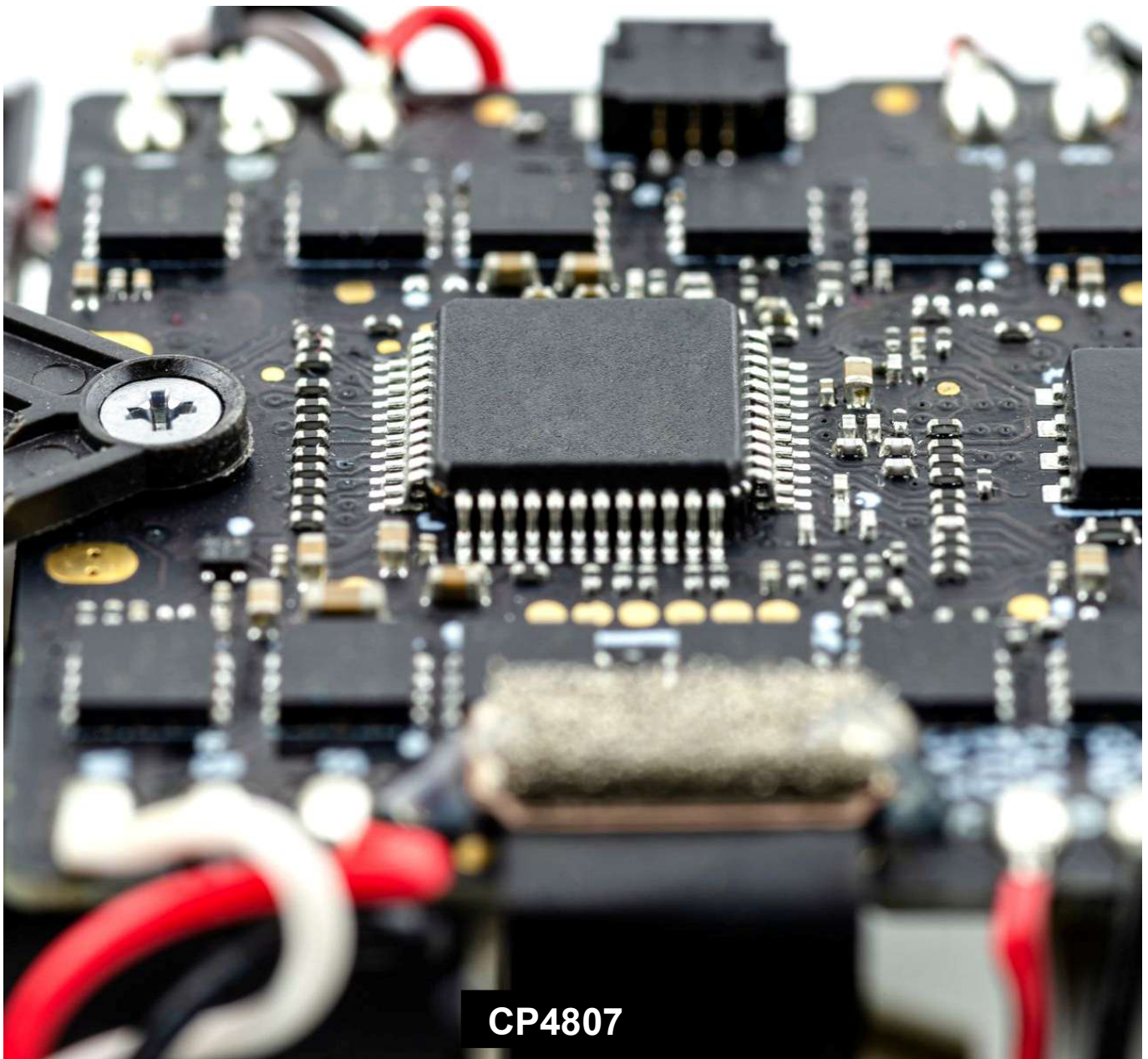


Introduction to microcontrollers



CP4807

Bronze

Worksheet 1	First program
Worksheet 2	Performing calculations
Worksheet 3	Connection points
Worksheet 4	Digital inputs
Worksheet 5	Making decisions
Worksheet 6	Macros / subroutines
Worksheet 7	Using prototype boards

Silver

Worksheet 8	Colour graphical displays
Worksheet 9	Pin interrupts
Worksheet 10	Timer interrupts

Gold

Worksheet 11	Touch control systems
Worksheet 12	Web mirror

Teacher's notes

Before tackling the worksheets you will need to understand some basic information:

Microcontroller basics

This document gives a basic explanation as to how microcontrollers work.

Introduction to E-blocks

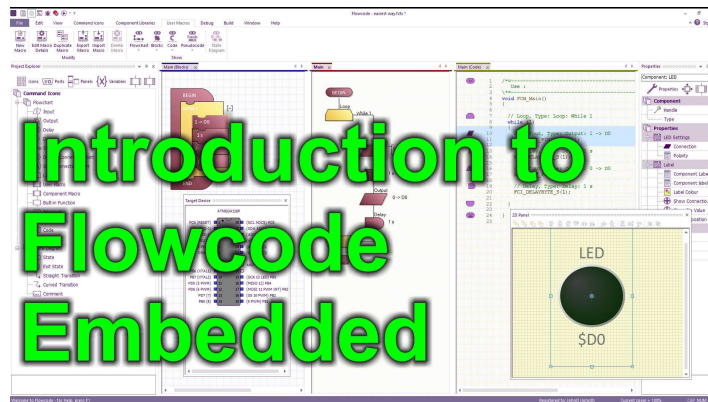
This document explains how the E-blocks system works.

E-blocks datasheet

This document gives the full schematics of all the E-blocks 3 circuit boards.

Flowcode Embedded getting started guide

This document shows you how to construct your first program from scratch.
Introduction to Flowcode Embedded video



<https://youtu.be/9WAcK9LjE>

IMPORTANT

Before you start using Flowcode make sure that you have the latest libraries by selecting HELP...LIBRARY UPDATES...FULL DATABASE (FILES IN USE DROP BOX)...DOWNLOAD.

Worksheet 1

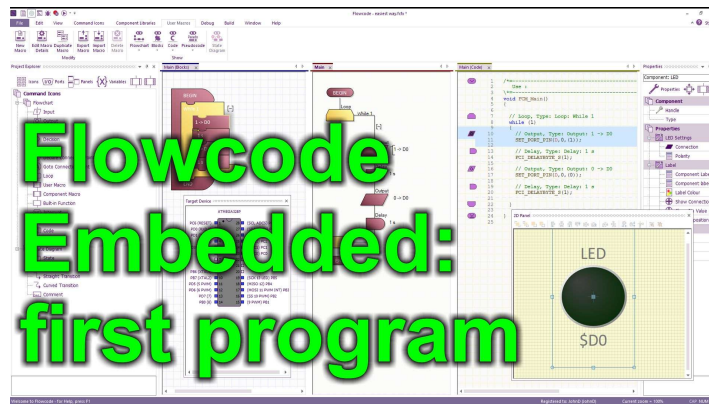
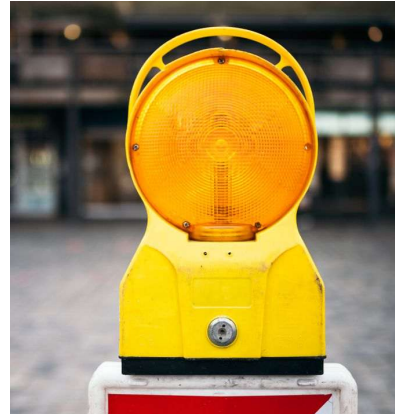
First program

CP4807-1

Introduction to microcontrollers

This first program introduces you to several programming commands: OUTPUT, LOOP and DELAY.

In this first program you explore how to get a single output on the microcontroller turning on and off using these commands.



<https://youtu.be/h8-7BsXBpLc>

Over to you:

- Watch the video “Flowcode Embedded: first program” on the Flowcode YouTube site.
- Load the file “First program”. Set up the hardware appropriately, download it, and make sure you understand it.
- The program flashes a single LED on the combo board.

Challenges:

- Delay: alter the delay period. Make sure you understand how the Delay command works.
- Output: change the output to work with port A and port B. Output: Select Advanced options and alter the bit that is flashed
- Loop: experiment with the loop dialogue box. Flash the LED for only 10 times.
- Alter the program so that it flashes Port 1 bit 0 for 10 times, then Port 2 bit 1 for 9 times then port 2 bit 2 for 8 times.

Hints:

1. You can use three loops one after the other. Alter the Loop count in each one.
2. Use the Advanced options in the Output icon to control which output bit is used.
3. The examples are available directly from within the Flowcode package. From within Flowcode select: FILE...NEW PROJECT...OPEN EXAMPLE OR TEMPLATE...and select the example you want from

the relevant worksheet pack.

Worksheet 2

Performing calculations

CP4807-2

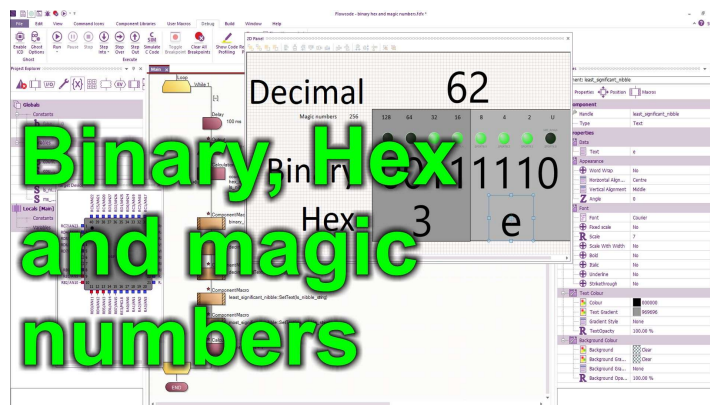
Introduction to microcontrollers

Microcontroller brains are not wired like human beings. This has consequences for mathematics, variables and input output processes that you need to understand.

In this worksheet you explore calculations and binary and hexadecimal numbering systems.



Part 1



<https://youtu.be/iEkocOsMj6Q>

Over to you:

- Watch the video “Binary, Hex and Magic numbers” on the Flowcode YouTube site.
- Load the file “Binary Hex and magic numbers”. This is a simulation only program.
- There are some commands in here that you will not yet understand. Don’t worry about that for now: just look at the results of the simulation.

Challenges:

- Simulate the program and pause it or slow it down to help you understand.
- Make sure you understand conversion between Binary, Hex and Decimal.
- Test yourself by selecting 3 decimal numbers between 0 and 100 and converting them to binary
- Repeat the exercise but convert 3 binary 8 bit numbers into decimal.

If you are using an ESP32 microcontroller this worksheet will present difficulties as the ESP32 is a 32 bit system and chip input output connections are complex.



Worksheet 3

Using connection points

CP4807-3

Introduction to microcontrollers

There are always multiple ways of making a program. Sometimes this is down to personal preference, sometimes this is down to good programming technique.

In this exercise you explore two ways of making a program: using connection points and loops.



There is no video for this worksheet.

Over to you:

- Load the results of the previous worksheet. You should have a program that lights up the LEDs on a port in sequence going from bit 0 to 7 and then bit 7 to 0.
- Use connection points to make the program run forever.

Challenges:

- Using connection points start from scratch and create a traffic light program that forever loops between 'Red' for 10 seconds, 'Amber' for one second, 'Green' for 10 seconds, 'Amber' for 1 second. 'Red' for 10 seconds and so on.
- Repeat the exercise without using any connection points – just use Loops.
- Which is the easiest program to understand?

Hints:

1. Add a connection point at the start of the program.
2. At the end of the program add a Goto Connection point.
3. Traffic lights: use a single loop with Output commands and delays to achieve the result.

Worksheet 4

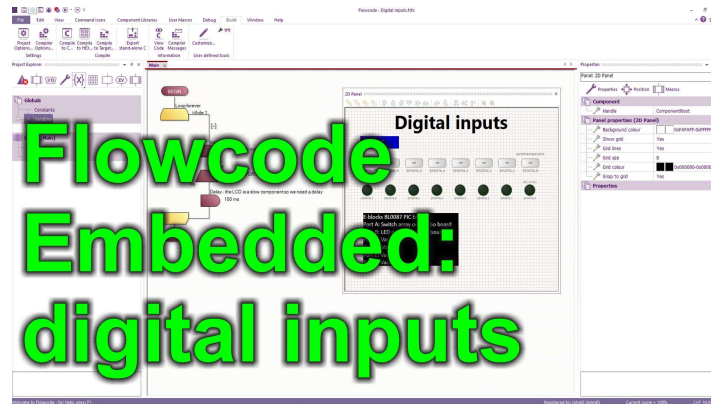
Digital inputs

CP4807-4

Introduction to microcontrollers

We need a way of controlling electronic devices locally. Often this is done in a digital context – with different types of switches. In this case we need to get the input ‘commands’ and then process this to alter the state of the microcontroller or alter its processes.

In this exercise you look at how to get command data into a microcontroller.



<https://youtu.be/Xvs7-iTPXcQ>

Over to you:

- Watch the video “Flowcode Embedded: digital inputs” on the Flowcode YouTube site.
- Load the file “Digital inputs”. Set up the hardware appropriately, download it, and make sure you understand it.
- The program reflects the input on one 8 bit port to another by storing the port status into a variable and then sending that to the output of another port.

Challenges:

- Alter the program so that it works on only the least significant input bit using a Boolean variable.
- Alter the program so that the state of the least significant bit input is reflected to the most significant bit of the same port.

Hints:

1. Setup a new variable called ‘input_bit0’ of Type Boolean.
2. Use the Advanced options of the Input command to read the state of the switch on bit 0 into ‘input_bit0’.
3. Change the Output icon using Advanced options so that it outputs ‘input_bit0’ to bit 7 of the same port as the input: so now you have an output LED on bit7 and an input switch on bit 0.

Challenges:

- Alter the program again so that the state of the second bit input switch is placed in a Boolean variable 'input_bit1'
- Alter the program so that bit 7 of the same port only lights up when both 'input_bit0' and 'input_bit1' are logic high – or 1.

Hints:

1. Use the AND operator in the Output statement: input_bit0 AND input_bit1

Challenges:

- Look up Mathematical operators in the Flowcode Wiki: select HELP...WIKI
- Experiment with operators OR, XOR, and NOT and make sure you understand how they are used.

Worksheet 5

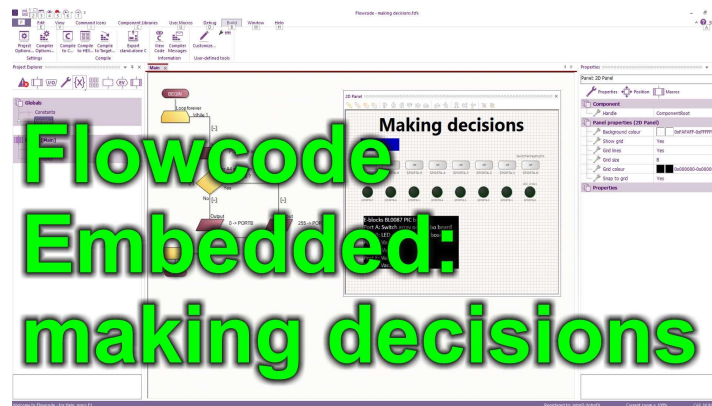
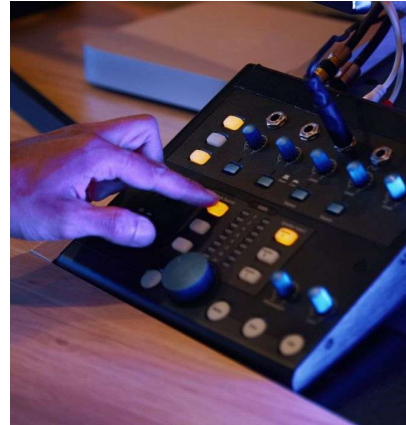
Making decisions

CP4807-5

Introduction to microcontrollers

The behaviour of a program and its processes will often depend on the choices made by the user via a control panel or input from sensors. The way we make decisions in a program is through the IF statement and the SWITCH statement.

In this worksheet you explore how decisions are made in microcontroller programs.



<https://youtu.be/jBTPa4wSFsl>

Over to you:

- Watch the video “Flowcode Embedded: making decisions” on the Flowcode YouTube site.
- Load the file “Making decisions”. Set up the hardware appropriately, download it, and make sure you understand it.

Challenges:

- Alter the program so that when switch A.0 and switch A.4 are pressed all the LEDs on port B are lit.
- Use a SWITCH command icon with 8 branches so that when the switch on A.0 is pressed the LED on B.7 is lit, when switch A.1 is pressed the LED on B.6 is lit etc.

Hints:

1. Use your knowledge of binary to work out the input when A0 AND A4 are pressed.
2. Write down the binary input for each switch press
3. Set up a SWITCH statement with 8 branches which tests for each switch press

Worksheet 6

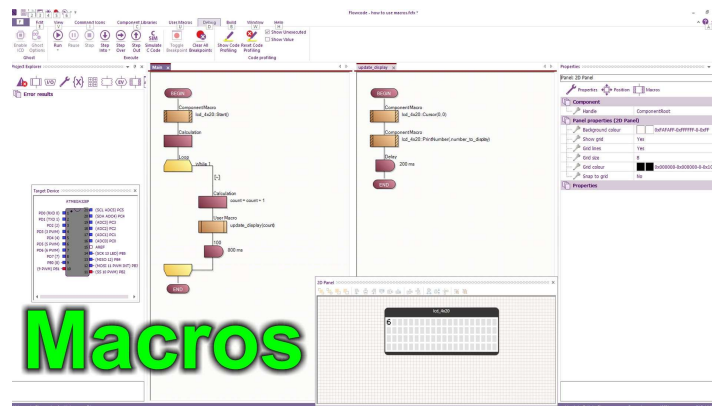
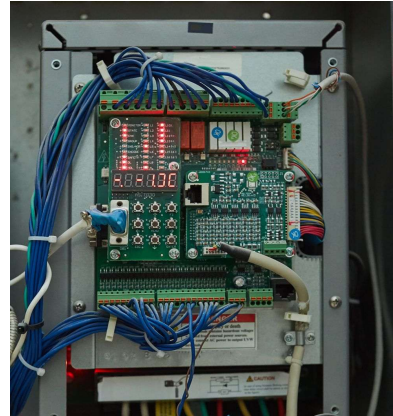
Macros / subroutines

CP4807-6

Introduction to microcontrollers

In designing microcontroller systems we use code that is very repetitive and we can simplify the development by using software macros or subroutines. Pre-written libraries of code for hardware modules also save us a lot of time, and help us create more functional systems.

In this worksheet you explore the use of hardware and software macros.



<https://youtu.be/hNBSxdb72v4>

Over to you:

- Watch the video “Macros” on the Flowcode YouTube site.
- Load the file “how to use macros”. Set up the hardware appropriately, download it, and make sure you understand it.

Challenges:

- Alter the program so that it Prints “Count is: “ then the count number
- Alter the program so that it prints the number on the second line of the display rather than the first
- Assign the text “Count is: “ to a variable and use the string variable in your program.
- Add a second software macro to the program called “Initialise”.
- Move the Start command to this macro and then call it from the Main routine.

Hints:

1. Use the hardware macro PrintString to print a string (you need to use inverted commas in the dialogue box)
2. Add a new variable “mystring” of Type String and length 20. Use PrintString to print this.
3. Use the Cursor hardware macro to alter the print position.

Challenges:

- Develop a program that uses the 7-segment displays on the Combo board to display the value of Count.
- Create a document that tells someone new to your project how it works.

Hints:

1. Use a BL0114 Combo board component from the Hardware group and the Seg_Showdigit macro
2. Use three variables: Units, Tens Hundreds.
3. Use some maths to convert Count into Units, Tens and Hundreds - Investigate the MOD mathematical function – see the Wiki for details.
4. Use the FILE....EXPORT...as HTML option to create a HTML document of your project. There is a lot of information here. Identify the parts you want and use copy and paste to assemble a report on what you have done.

Worksheet 7

Using prototype boards

CP4807-7

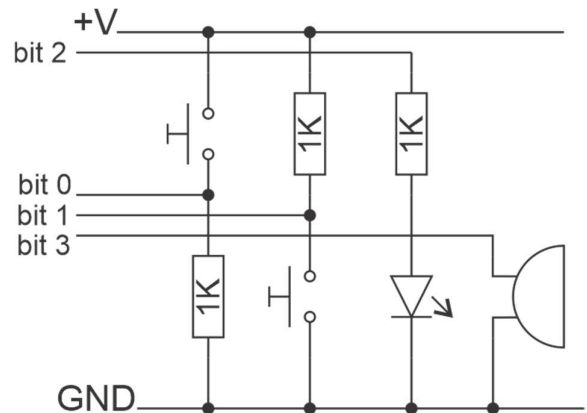
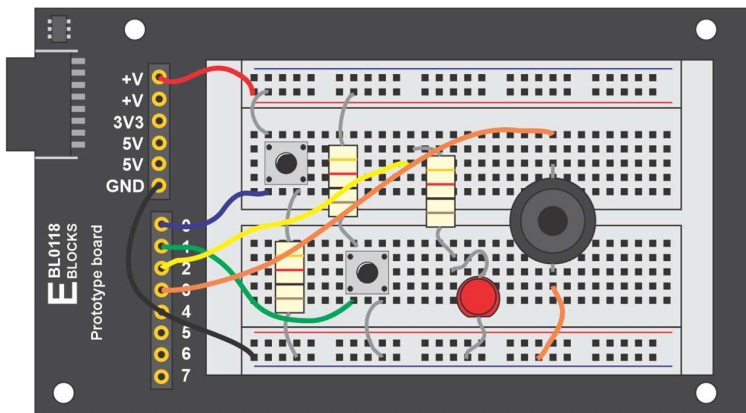
Introduction to microcontrollers

So far you have developed programs using boards which are manufactured for you. At some point you will need to create your own circuits. You can do this with a prototype board. This links to E-Blocks very easily to provide you with the connections and power supplies that you need.

In this worksheet you explore the use of prototype boards in microcontroller circuits.



There is no video for this worksheet.



Over to you:

- The diagrams above show you a circuit that you need to build. Two switches (active high and active low), a single LED and a buzzer.
- Build this circuit and add it to your upstream E-blocks system on any port.
- Develop a program that:
 - When both switches are pressed activates the buzzer.
 - Flashes the LED once a second

Hints:

1. You can start from scratch or start from a previous program – perhaps the one on Digital inputs
2. The main problem here is that there are two functions which need to be carried out at the same time. It is difficult to do both at once. (later you will learn how to use interrupts for this situation). Set up a loop with a 10ms delay. 100 of these loops gives you 1 second. Set up a variable to track a second has elapsed. Use this to change the state of the LED output.
3. Use the && operator for the logic of both switches being pressed at once. One of your switches is active high and the other active low.

Worksheet 8

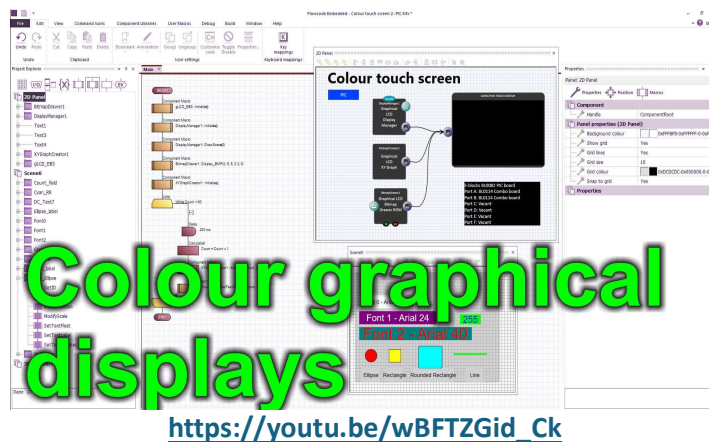
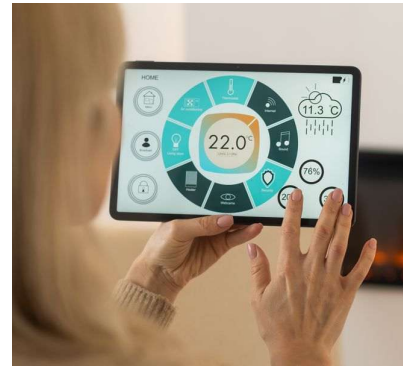
Colour graphical displays

CP4807-8

Introduction to microcontrollers

Graphical displays extend the functionality of electronic systems, allow more information about be displayed, and allow information to be displayed in more easily understood formats: tables, graphics, graphs etc.

In this worksheet you explore how colour graphical displays work.



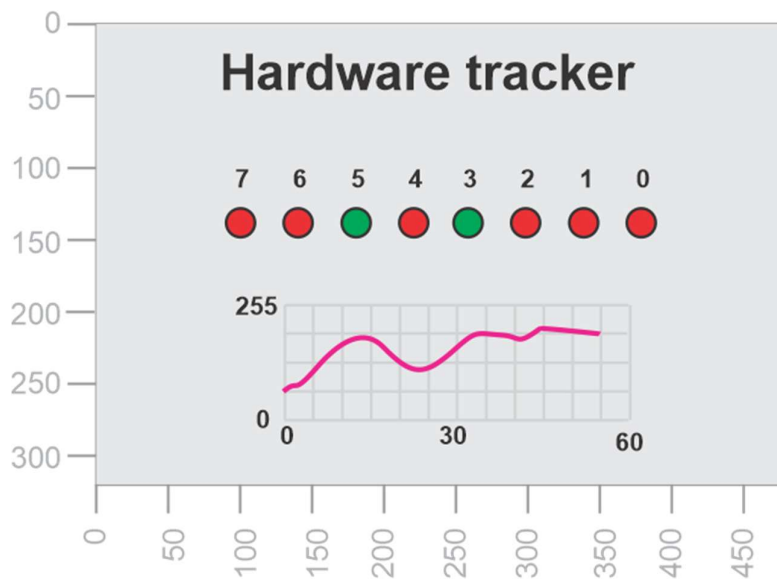
https://youtu.be/wBFTZGid_Ck

Over to you:

- Watch the video “Colour graphical displays” on the Flowcode YouTube site.
- Load the program “Colour graphical displays”. Set up the hardware appropriately, download it, and make sure you understand how the program works.

Challenges:

- Create a program that uses the colour graphical display to show the status of 8 of the switches on the Combo board and the value of a potentiometer on an XY graph.
- Your display will look something like this:



Hints:

1. Use 8 ellipses to show the state of the switches. Each ellipse has a ModifyColour macro you can use to change ellipse colour – use red (255,0,0) for switch off and green (0, 255, 0) for switch on.
2. Use the graph component to show the value of the potentiometer. You may need to complete the first worksheet in Sensors and Microcontrollers to understand how to do get a value from a potentiometer.
3. Use text fields appropriately to label your components.
4. Modify colours on all components to make your screen look great.
5. If you can; create a bitmap using a graphics package and use the Bitmap drawer component to display it.
6. Refresh the graph every 60 seconds – you don't need to clear the whole display.

Worksheet 9

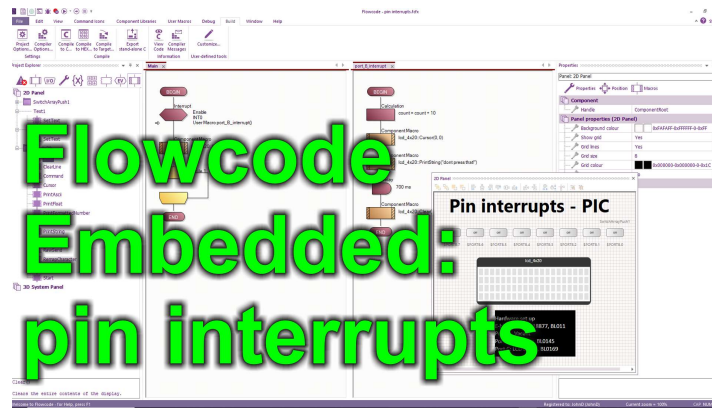
Pin interrupts

CP4807-9

Introduction to microcontrollers

In some systems there are possible events that are really important – like the kill switch on a factory machine. When switches like this are activated you need your program to stop whatever it is doing – ‘interrupt’ the program - and take action. Interrupts allow us to do this.

In this worksheet you explore how pin interrupts work.



<https://youtu.be/imTq19jX408>

Over to you:

- Watch the video “Flowcode Embedded: pin interrupts” on the Flowcode YouTube site.
- Load the file “pin interrupts”. Set up the hardware appropriately, download it, and make sure you understand it.

Challenges:

- Alter the program so that a LED output flashes during normal operation.
- Alter the program so that once the ‘stop switch’ on the interrupt pin is activated the program waits until a ‘clear’ switch is pushed before continuing the program.

Hints:

1. Put a flashing LED routine in the main loop.
2. In the Interrupt routine insert a While...until loop.
3. Use a variable to track the status of the ‘clear’ switch.
4. In the loop test for the ‘clear’ switch being pressed.

Worksheet 10

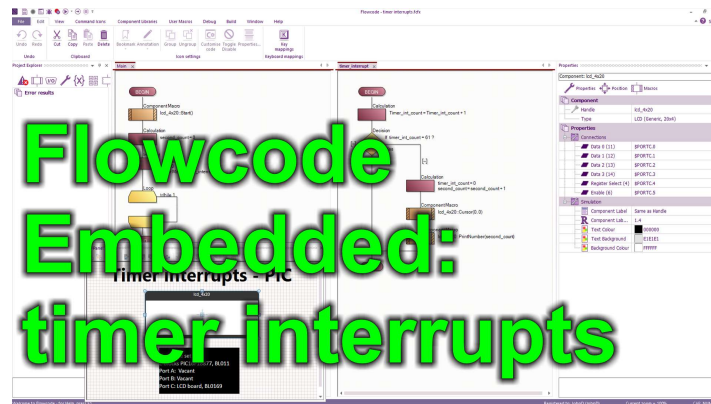
Timer interrupts

CP4807-10

Introduction to microcontrollers

Many microcontroller systems need some kind of timing system. Microcontrollers are designed to help us here as they have dedicated internal hardware that counts the number of internal clock pulses and converts them into time. When activated these internal counters can 'interrupt' the program to help you manage time-based code.

In this worksheet you explore how timer interrupts work.



<https://youtu.be/E5SWKxIjvno>

Over to you:

- Watch the video “Flowcode Embedded: timer interrupts” on the Flowcode YouTube site.
- Load the file “timer interrupts”. Set up the hardware appropriately.

Challenges:

- Alter the program so that it calculates the hours, minutes and seconds that have elapsed.
- Use a switch and a pin interrupt to clear the timer.

Hints:

1. Create variables hour_count and minute_count
2. In the main loop use IF statements to track when second_count is more then 60 and reset it to 0 and increment minute_count Similarly for hour_count.
3. Keep the programming in the interrupt routine to a minimum – it is good practice
4. Create a new routine for a pin interrupt and use the code from the previous exercise to create the reset function.

Worksheet 11

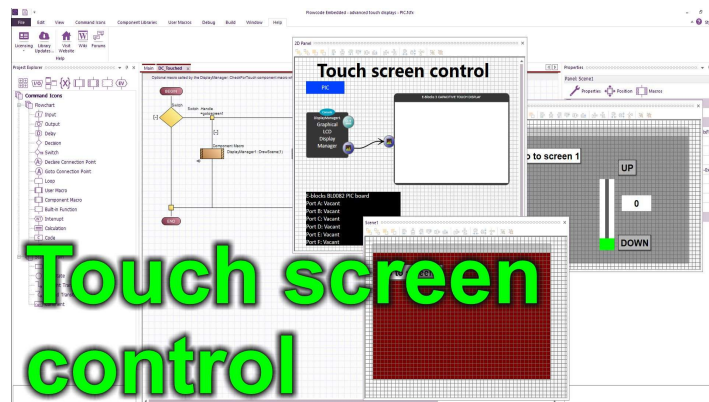
Touch control systems

CP4807-11

Introduction to microcontrollers

Touchscreen systems are replacing the use of switches, potentiometers, LEDs and other electronic devices. Using a graphical touch screen reduces costs and enhances functionality.

In this worksheet you learn how to develop a system that uses the touch screen display to control a simple system.



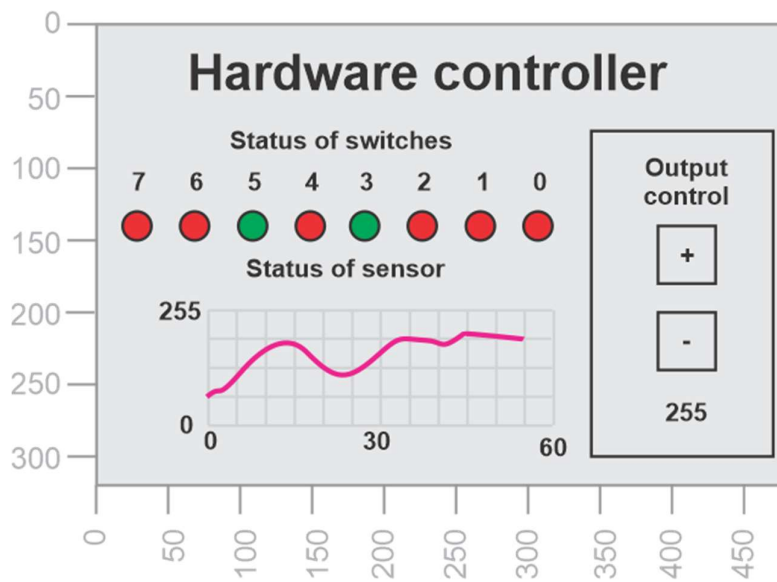
<https://youtu.be/vS7P8RzybKc>

Over to you:

- Watch the video “Touch screen control” on the Flowcode YouTube site.
- Load the file “Touch screen control”. Set up the hardware appropriately.
- Download the file and make sure you understand how the program works.

Challenges:

- Take the program you wrote in the Colour Graphical Displays program.
- Modify it so that it has two touch buttons: ‘+’ and ‘-’ and a text field
- Use these to control an output variable and show this variable (between 0 and 255) on a port of 8 LEDs.
- Your display will look something like this:



Hints:

5. Use a new variable: Output
6. Use the touch screen routine to detect when the '+' and '-' text fields are touched and modify the variable Output accordingly
7. Write the Output variable to a port to show its value.

Worksheet 12

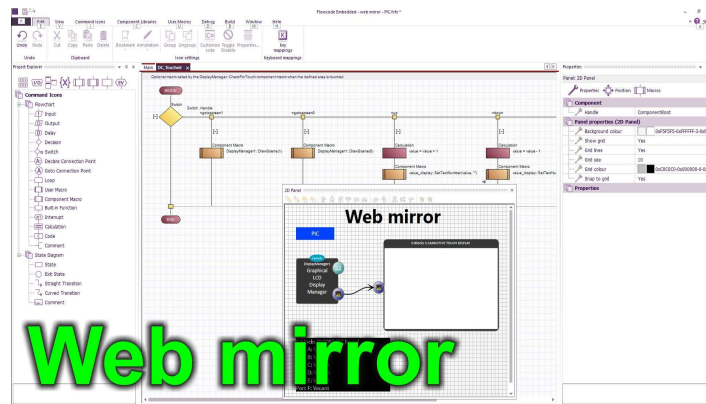
Web mirror

CP4807-12

Introduction to microcontrollers

Touchscreen systems are replacing the use of switches, potentiometers, LEDs and other electronic devices. Using a graphical touch screen reduces costs and enhances functionality.

In this worksheet you learn how to develop a more complex touch screen system



<https://youtu.be/rdSOYW6AXn0>

Over to you:

- Watch the video “Web mirror” on the Flowcode YouTube site.
- Load the file “Web mirror”. Set up the hardware appropriately. Change the EB3 Web Mirror settings of the Touch screen component and enter the name and password of your local server.
- Download the file and make sure you understand how the program works.
- Use your mobile phone camera to grab the URL of the remote server from the displayed QR code that allows you to control the system over the internet.

Challenges:

- Take the program you wrote in the Touch control systems worksheet.
- Change the EB3 Web Mirror settings of the Touch screen component to ‘Yes’ and enter the name and password of your local server.
- Compile the program to hardware.
- Make sure that you can use your program on your mobile phone to control and monitor your hardware.

Previous learning

Some understanding of electronics is needed.

Pedagogy

This set of worksheets provides structure for the study of microcontroller programming. The worksheets provide structure to the learning but do not provide all the information students need. Our vision as to how you teach this subject is:

1. This is a very practical subject. We expect that 90% of the time students spend on this module will be spent working alone.
2. Each student should have Flowcode Embedded software and a set of E-blocks 3 hardware. Don't be tempted to try to get students to learn this subject in pairs.
3. Students should be introduced to each topic or worksheet. They should be given time to investigate the topic using the internet. If students are investigating this on their own, you may want to give additional structure to this activity.
4. Students should then load an example program and download this to the microcontroller so that they can see how the program works. Example programs are available on the Flowcode Wiki for each of PIC, Arduino Mega and ESP32 microcontrollers.
5. Students should be given the opportunity to watch the accompanying YouTube video – individually or as a class activity. This explains the program.
6. Students then modify each example as detailed in the worksheet. This ensures that they have success, but also ensures that they have understanding.
7. Occasionally you should test students understanding by asking them to complete a project from scratch. This tests their understanding of what they have learned.
8. Students do not have to do all worksheets or tasks. That's up to you to decide. Worksheet 10 and 11 are quite advanced.
9. There are three sets of worksheets in this series: Introduction to microcontrollers, Sensors and microcontrollers, and Motors and microcontrollers. We have split the content by topic for ease of publication. You will need to decide what sequence your students complete the worksheets in. We have split the worksheet difficulty into Bronze, Silver, and Gold. Not all students will be able to complete all exercises. A reasonable approach would be to deliver Bronze in each of Introduction to microcontrollers, Sensors and microcontrollers, and Motors and microcontrollers, then deliver Silver then deliver Gold to those students who are capable.
10. The curriculum is based on the requirements of the BTEC level 3 in Engineering – Unit 6: Microcontroller systems for engineers. This can be downloaded from the Pearson web site.

Example programs

In general the worksheets make use of prewritten functional examples. Students load the example, understand it and then modify it to make sure that they have learned the principles under discussion.

The examples are available directly from within the Flowcode package. From within Flowcode select: FILE...NEW PROJECT...OPEN EXAMPLE OR TEMPLATE...and select the example you want from the relevant worksheet pack.

Examples are available separately for PIC, Arduino and ESP32 microcontrollers.

Time

Of course this depends on student capability. We estimate that it will take 15 hours to complete all the worksheets in this module.

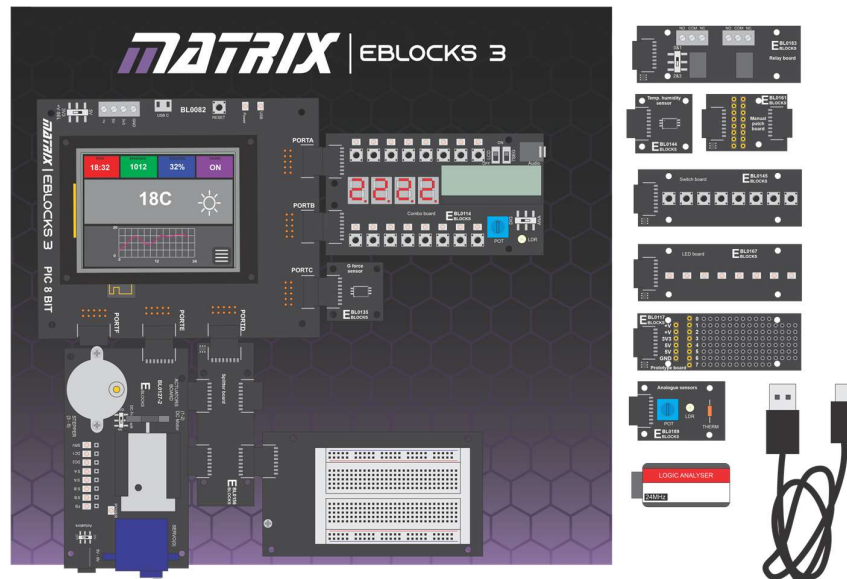
Software

The videos make use of Flowcode embedded and we anticipate that you are likely to use this for your programs. But you can use these worksheets with any microcontroller development software. The flowchart programs can easily be followed by C programmers.

Assessment

Creating programs from scratch will be suitable as a way of assessing that students have understood the programs and have programming ability.

Hardware



This course is designed to be used with a Matrix Microcontroller development centre. There are three versions:

- | | |
|--------|--|
| BL5394 | ESP32 microcontroller development centre |
| BL8624 | PIC microcontroller development centre |
| BL3797 | Arduino microcontroller development centre |

These consist of:

- Upstream board – BL0082 PIC, BL0040 Arduino, or BL0070 ESP32
- BL0114 Combo board
- BL0117 Prototype board
- BL0118 Project board
- BL0127 Actuators board
- BL0135 9 axis motion / accelerometer board
- BL0144 Temp/humidity board
- BL0145 Switch board
- BL0156 Splitter board
- BL0172 Logic analyser with ribbon cable
- BL0167 LED board
- BL0183 Relay board
- BL0189 Analogue board

BTEC National Unit 6 – topic coverage

A Investigate typical microcontroller system hardware		Covered?	Where covered
A1 Control hardware			
I/O capabilities – number, type (analogue/digital), ports		✓	All
hardware specification – bus width, processor speed		✓	All
memory – RAM, ROM		✓	All
hardware features – interrupts, stack, PWM		✓	All
required peripherals		✓	All
cost and accessibility		•	
ease of use		•	
software and programming language		✓	All
operating voltages and power requirements		✓	All
A2 Input devices			
User input:			
	digital – switches and buttons	✓	Introduction to microcontrollers
	analogue – control potentiometer	✓	Introduction to microcontrollers
Temperature			
	Thermistor	✓	Sensors and microcontrollers
	temperature sensors	✓	Sensors and microcontrollers
	environmental sensor – temperature and humidity	✓	Sensors and microcontrollers
Light			
	light-dependent resistor (LDR)	*	
	IR – phototransistor, photodiode or IR receiver	✓	Sensors and microcontrollers
Movement/orientation			
	tilt switch	*	Introduction to microcontrollers
Presence			
	micro-switch	*	Introduction to microcontrollers
	Ultrasonic	*	Introduction to microcontrollers
Input interfacing requirements			
	signal conditioning	*	
	analogue-to-digital (ADC) conversion	✓	Sensors and microcontrollers
	modular sensor boards	✓	Sensors and microcontrollers
	PWM	✓	Motors and microcontrollers
	serial communications	✓	Sensors and microcontrollers
	Inter-Integrated Circuit (I2C)	✓	Sensors and microcontrollers
A3 Output devices			
Optoelectronic			
	light-emitting diode (LED) – indicator and IR	✓	Introduction to microcontrollers
	7-segment display	✓	Introduction to microcontrollers
	liquid crystal display (LCD)	✓	Introduction to microcontrollers
Electromechanical			
	Relay		Introduction to microcontrollers
	direct current motor	✓	Motors and microcontrollers
	Servo	✓	Motors and microcontrollers
Audio			
	buzzer or siren	✓	Introduction to microcontrollers
	speaker or piezo transducer		
Output interfacing requirements			
	power requirements and drivers		
	transistor output stage		
	Relay		
	PWM	✓	Motors and microcontrollers
	serial communications	✓	Sensors and microcontrollers
	I2C device interfacing	✓	Sensors and microcontrollers
A4 Selecting hardware devices and system design			
A5 Assembling and operating a microcontroller system			

A Investigate typical microcontroller system hardware		Covered?	
A1 Control hardware			
I/O capabilities – number, type (analogue/digital), ports		✓	Introduction to microcontrollers
hardware specification – bus width, processor speed		✓	All
memory – RAM, ROM		✓	All
hardware features – interrupts, stack, PWM		✓	Introduction to microcontrollers
required peripherals		✓	Introduction to microcontrollers
cost and accessibility		•	
ease of use		•	
software and programming language		✓	All
operating voltages and power requirements		✓	Introduction to microcontrollers
A2 Input devices			
User input:			
	digital – switches and buttons	✓	Introduction to microcontrollers
	analogue – control potentiometer	✓	Sensors and microcontrollers
Temperature			
	Thermistor	✓	Sensors and microcontrollers
	temperature sensors	✓	Sensors and microcontrollers
	environmental sensor – temperature and humidity	✓	Sensors and microcontrollers
Light			
	light-dependent resistor (LDR)	✓	Sensors and microcontrollers
	IR – phototransistor, photodiode or IR receiver	*	
Movement/orientation			
	tilt switch	*	Project
Presence			
	micro-switch	*	Project
	Ultrasonic	*	
Input interfacing requirements			
	signal conditioning	*	
	analogue-to-digital (ADC) conversion	✓	Sensors and microcontrollers
	modular sensor boards	✓	Sensors and microcontrollers
	PWM	✓	Motors and microcontrollers
	serial communications	✓	Sensors and microcontrollers
	Inter-Integrated Circuit (I2C)	✓	Sensors and microcontrollers
A3 Output devices			
Optoelectronic			
	light-emitting diode (LED) – indicator and IR	✓	Introduction to microcontrollers
	7-segment display	✓	Introduction to microcontrollers
	liquid crystal display (LCD)	✓	Introduction to microcontrollers
Electromechanical			
	Relay	✓	Introduction to microcontrollers
	direct current motor	✓	Motors and microcontrollers
	Servo	✓	Motors and microcontrollers
Audio			
	buzzer or siren	✓	Introduction to microcontrollers
	speaker or piezo transducer		Introduction to microcontrollers
Output interfacing requirements			
	power requirements and drivers	✓	Introduction to microcontrollers
		transistor output stage	Project
		Relay	Project
	PWM	✓	Motors and microcontrollers
	serial communications		Sensors and microcontrollers
	I2C device interfacing		Sensors and microcontrollers
A4 Selecting hardware devices and system design		✓	All
A5 Assembling and operating a microcontroller system		✓	All
B1 Programming techniques		✓	All
B2 Coding constructs		✓	All
B3 Number systems		✓	Introduction to microcontrollers
C1 Development processes		✓	All
C2 Documentation		✓	Introduction to microcontrollers

