

Using Flowcode PC Developer to Communicate with Embedded Projects

Introduction

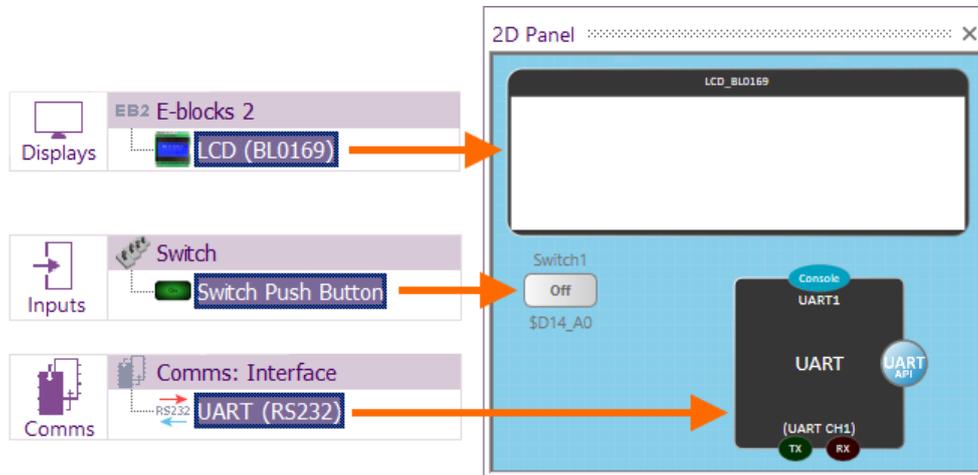
This worked example will show how to use Flowcode to create an embedded project that communicates with a PC via the serial port. A PC app will also be developed in Flowcode using PC Developer.

This basic example will show a simple Arduino-based app that checks for incoming serial port data and prints a message on a display if anything is received. An associated PC app will be developed to send messages to the embedded app via the connected COM port. These two apps will also show how to send serial messages from the embedded app to the PC app.

The hardware will comprise of an Arduino Uno board connected to an LCD to display received messages and a switch to send messages to the PC.

Embedded App

The goal of this basic example is to send and receive serial data, which requires the use of a UART (RS232) component. An LCD component is needed to display received messages and a button switch is used to send messages from the embedded app. These components will the following components need to be placed onto a component 2D panel:



Drag each of these components onto a 2D panel and then set their properties as shown below:

Component	
Handle	LCD_BL0169
Type	LCD
Properties	
Data 0 (11 / 7)	\$PORTB.0 / \$D8
Data 1 (12 / 8)	\$PORTB.1 / \$D9
Data 2 (13 / 9)	\$PORTB.2 / \$D10
Data 3 (14 / 10)	\$PORTB.3 / \$D11
Register Select (4)	\$PORTB.4 / \$D12
Enable (6)	\$PORTB.5 / \$D13

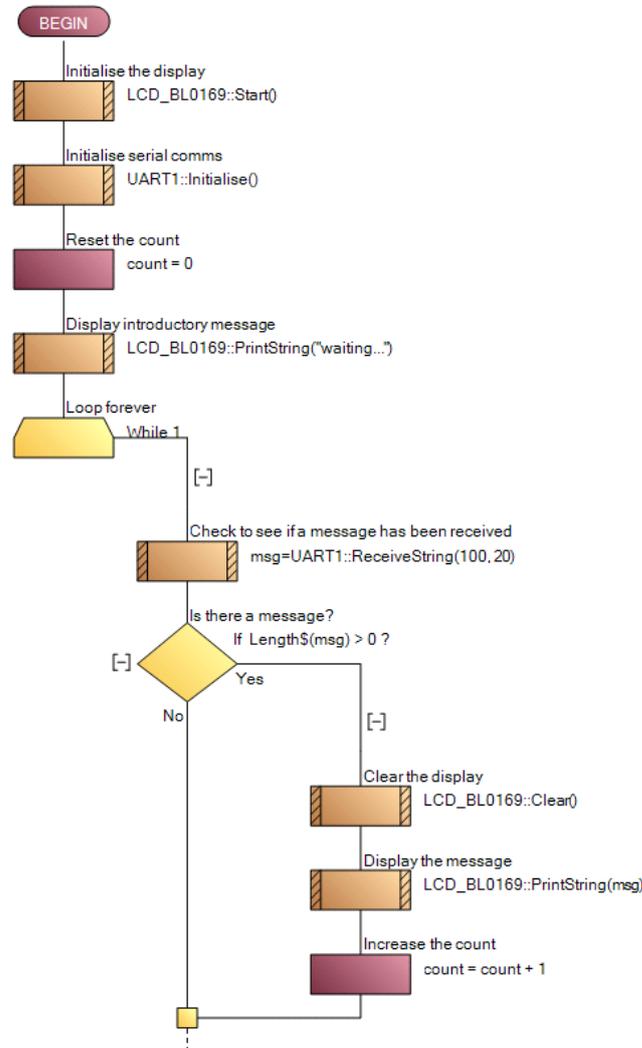
Component	
Handle	UART1
Type	UART (RS232)
Properties	
Comms Settings	
Channel	Channel 1
Baud Options	9600
Baud Rate	9600
Data Bits	8 Bits
Return	16 Bits
Echo	Off
Connections	
Use TX	Yes
TX	\$PORTD.1
Use RX	Yes
RX	\$PORTD.0
Use Flow Con...	Off

Component	
Handle	Switch1
Type	Switch Push Button
Properties	
Switch Settings	
Connection	\$D14_A0
Polarity	Active High
Debounce	5
Latching	Yes
Label	
Component L...	Same as Handle
Component L...	Top
Label Colour	787878
Show Connec...	Yes
Show Pin Value	No
Pin label position	Bottom
Appearance	
Style	Blue
Type	Rounded (wide)
On Label	On
Off Label	Off

The embedded app will have just a single “Main” macro and use the following global variables:

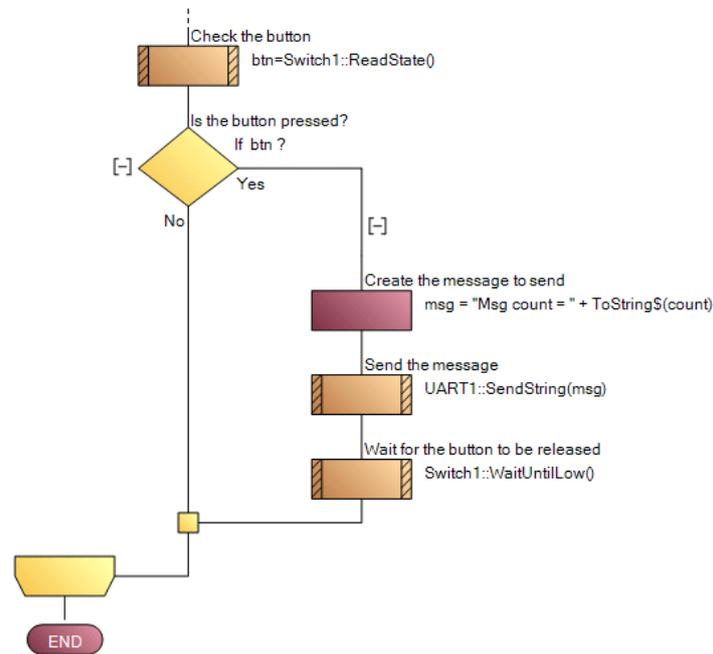
Variable	Type	Purpose
count	Integer	Keeps track of the number of received messages
btn	Byte	Reads the state of the switch to detect when it is pressed
msg[20]	String[20]	Contains the messages received and to be sent

The first half of this macro is shown below:



The first few icons initialise the display and the serial comms components. Following this, an integer variable – ‘count’ – is set to zero and then an introductory message is displayed. The remainder of the Main macro consists of an infinite loop which continually checks for incoming data on the serial port. If any data is detected, the ‘msg’ string variable will be populated with the message and will therefore have a size greater than zero. In this is the case, the LCD is cleared and the message is displayed. The ‘count’ variable increments by one each time an incoming message is detected and displayed.

The remainder of the infinite loop is shown below and enables the sending of messages to the PC. The state of the switch input is first read and if it is currently pressed, a message is created which uses the 'count' variable to inform how many messages have been received. This message is sent via the UART component to the PC and then the program flow is halted until the switch state becomes low again – i.e. until the user has released the connected button.



Once this flowchart has been created, the program can be compiled and downloaded to the Arduino using the “Build...Compile to Target” menu item. Remember to specify the port that connects the PC to the Arduino on the “Project Options” screen. The Arduino port will be listed in the appropriate dropdown box:

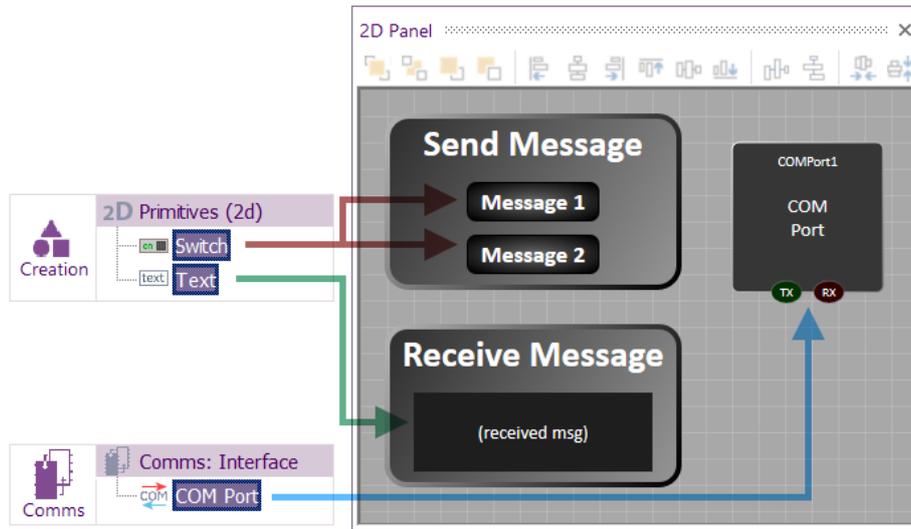
Programmer Port:

PC Developer App

The associated PC Developer app will behave in an equivalent way to the embedded app. It will check for incoming data and display it if any is found. It will also send text data out when a button is pressed.

The app will consist of two buttons send two different messages and a text field to display any received text. A “COMPort” component is also required, which allows data to be sent to and received from the PC’s serial port.

Here is the design of the basic web app:



Extra visual styling elements have been added to make the app look nice, but these are not necessary. The four essential components are shown and can be dragged onto a 2D panel from the indicated location in the Components menu of Flowcode.

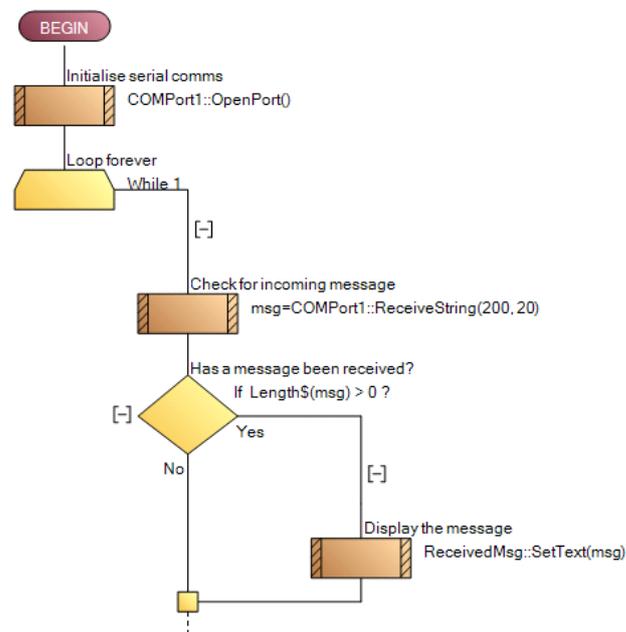
Details of the COMPort component is shown below. The “COM Port” entry need to be appropriate for your hardware connection. Also show below is the properties of one of the switches. The properties of the other switch will be the same (except with different label text). The Text component will have no unusual property settings and can retain its defaults.

Component	
Handle	COMPort1
Type	COM Port
Properties	
COM Port	COM4 Arduino Uno
Refresh COM Ports	No
Baud	9600
Timeout Value	256
Console Data	No
Auto Close	Yes
New Line Format	\r\n - CRLF

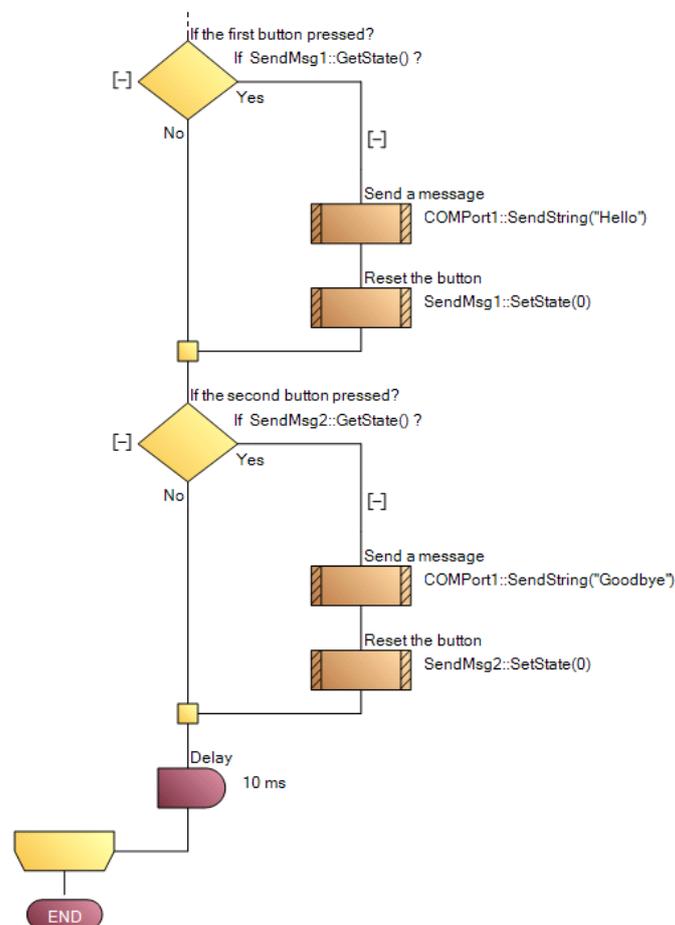
Component	
Handle	SendMsg1
Type	Switch
Properties	
Data	
State	Off
Macro Called	
Appearance	
Style	Custom Colours
Type	Rounded (wide)
Draw Thumb	No
Animation Time	0
Auto Return	No return (toggle)
Return Time	400
On Label	Message 1
Off Label	Message 1

A single global variable, “msg” is used to receive and display data. This is a string using the default array size [20].

As with the embedded app, there is a single “Main” macro which contains initialisation code followed by an infinite loop which checks for incoming data and for user button presses. The first part of this Main macro is shown below. Notice the check for an incoming message is like the embedded app:



The second half of the infinite loop checks each switch, and a message is sent if a switch has been pressed. After sending a message, the state of the switch is reset so it can be pressed again.



This app can be run from within Flowcode using the “Run” button in the “Debug” menu and if it is working correctly can be deployed as a stand-alone PC app using the File...Export menu.

Conclusion

This simple example shows two projects – an embedded app and a PC-based app – that communicate with each other using the serial port, exchanging text messages between the embedded device and the PC.

They have been developed within Flowcode using similar techniques with just a few components and flowchart commands and can form the basis of much more complex interactions between a PC and an external embedded device.