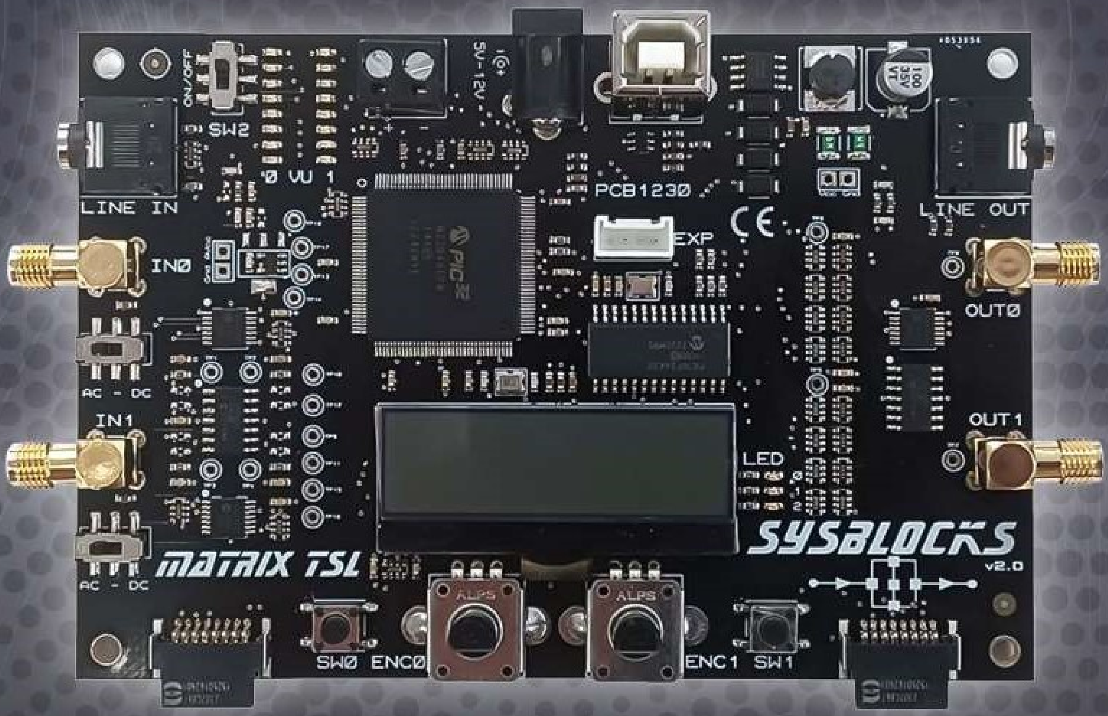
MATRIX

INTRODUCING

Sysblocks



SYSBLOCKS AND FLOWCODE 10

THE FUTURE OF SIGNAL PROCESSING TEACHING

Excellence in education for over 30 years

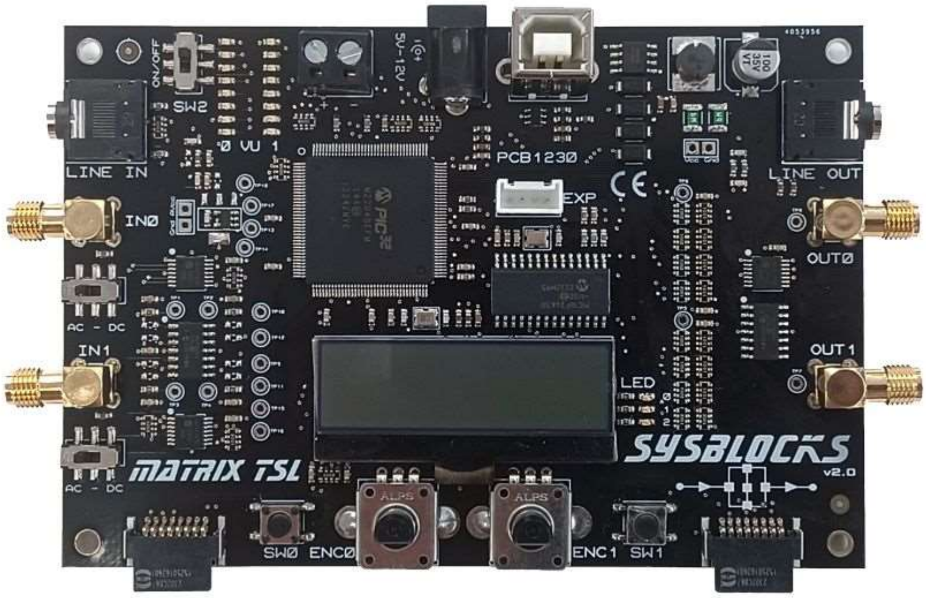
www.matrixtsl.com

Manufactured in the UK · MATRIX · 1993

BL4386

Contents

Introduction to Sysblocks



Sysblocks and Flowcode 10 provide a new way of teaching about signal processing for Music technology, DSP, Communications and Software Defined Radio.

Hardware:

The opportunity of developing a product like Sysblocks is facilitated by the advent of advent of low cost high speed, high power microcontrollers - in this case a 32bit PICmicro microcontroller. The Sysblock hardware topology is very simple: Input buffer …..A/D….processor...D/A….output amplifer.

Sysblocks boards also include two E-blocks II boards so that students can use a wide variety of expansion boards from Matrix, Grove, Mikroelektronika and others.

Software:

Flowcode 10 software is a graphical programming solution for microcontrollers and PCs. Flowcode allows those with limited programming experience to develop complex electronic systems using graphical and conventional C code programming.

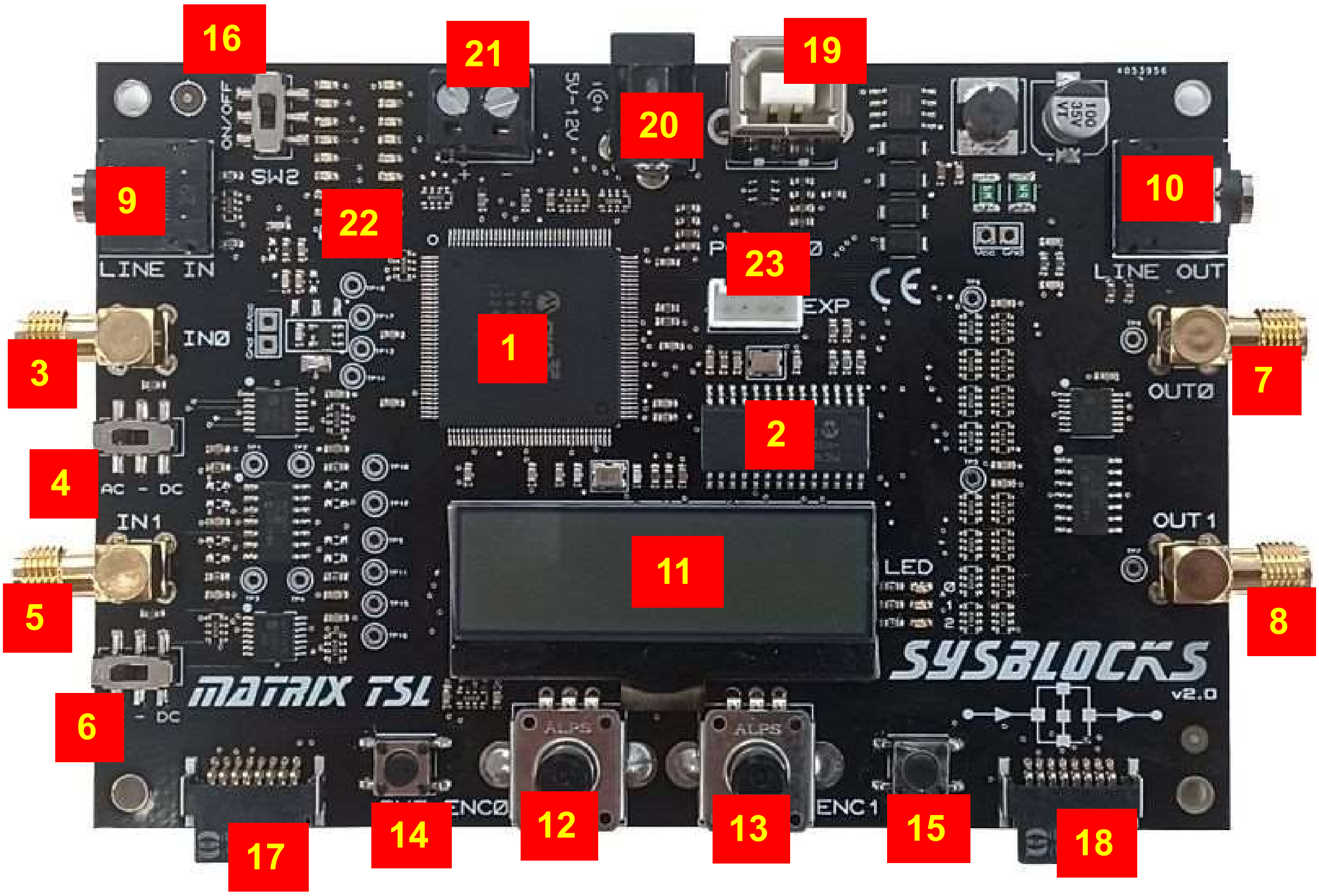
Curriculum:

There are three separate curricula for Sysblocks and Flowcode 10:

- Music with microcontrollers
- Systems, signals, DSP, and FFT
- Communications and Software Defined Radio

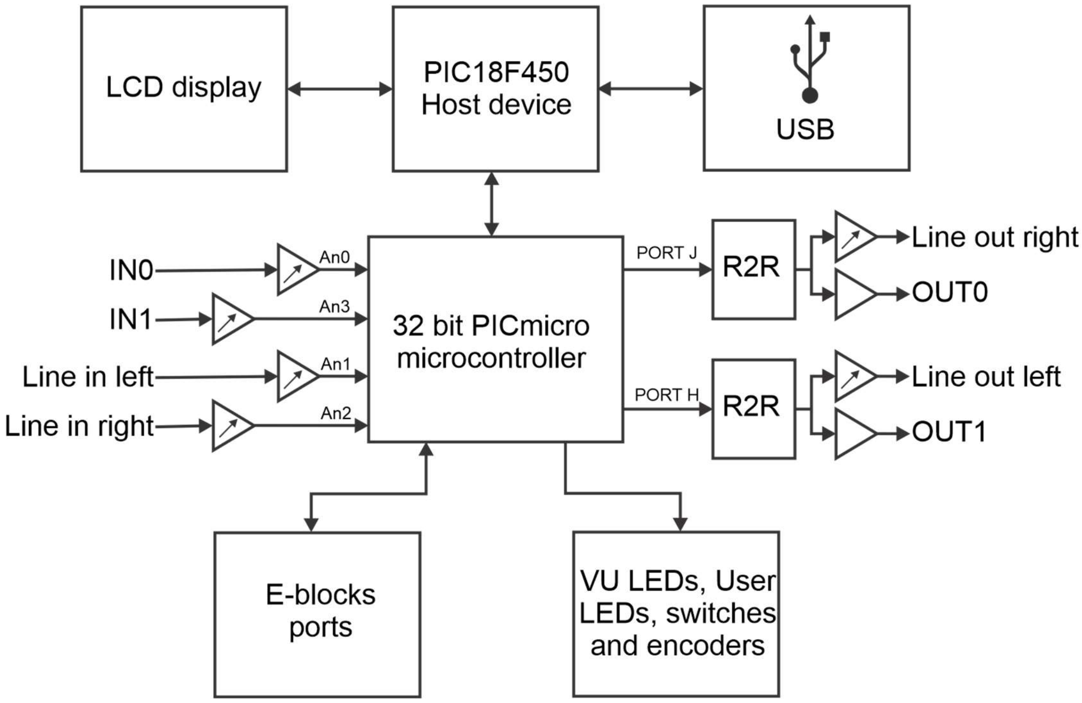
The following information will explain to you how Sysblocks and Flowcode 10 can be used to provide fantastic learning opportunities in electronics.

Hardware: PCB overview



1. Fast 32 bit PIC processor
2. PIC18F24K50 control processor that handles the USB communications, 32 bit PIC programming, LCD display, encoders and input switches
3. SMA connector Input IN0
4. AC / DC coupling switch SW4
5. SMA connector Input IN1
6. AC / DC coupling switch SW3
7. SMA connector Output OUT0
8. SMA connector Output OUT1
9. 3.5mm jack stereo line in
10. 3.5mm jack stereo line out
11. 4 line monochrome LCD display
12. Rotary encoder input ENC0
13. Rotary encoder input ENC1
14. Input switch SW0
15. Input switch SW1
16. Input switch SW2
17. E-blocks II expansion port A
18. E-blocks II expansion port B
19. USB socket / power in
20. 2.1mm power jack, 5 - 12V
21. Screw terminal power connector (input or output)
22. 2 x banks of 8 user programmable LEDs - for VU meters
23. Grove sensor expansion socket

Hardware: block diagram



Sysblocks has four analogue inputs, two analogue outputs and support circuitry.

At the heart of the system is a very powerful 32 bit Micro chip PICmicro microcontroller running at 200MHz. This is capable of standard microcontroller type functions and also is capable of Digital Signal Processing the incoming audio signals. The 32 bit processor has a Coremark of 652 (around 330 Drystone MIPS) which is more than fast enough for the signal processing tasks involved for learning about Music technology, Digital Signal Processing and Modern communications theory.

The 32 bit PICmicro microcontroller is programmed by a host device which is a standard 18 series PICmicro microcontroller. This host device takes care of non-core tasks like USB communications, LCD display, encoder and switch inputs. This architecture keeps the main 32bit device free for signal processing.

Two E-blocks ports are made available for expansion. E-block boards for Wifi, keypad, Bluetooth, CAN, Zigbee and many other functions are available. This gives huge flexibility to the Sysblocks system as a wide range of projects can be built around it. A Grove connector is also available which allows users to access the wide range of Grove accessories.

The analogue input signals are buffered and level shifted to half of the supply voltage – 1.65V – and then fed to the internal A to D converters of the main 32 bit PICmicro microcontroller. A digital potentiometer allows the level of the input signals to be adjusted in software.

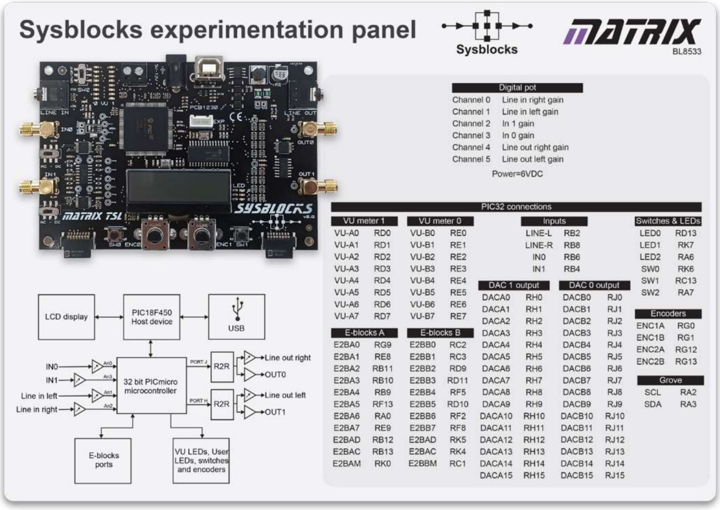
The microcontroller processes the digital signals and feeds the results to 2 x external R2R ladder DACs which are buffered by operational amplifier circuits. Line out left and right include a digital potentiometer that allows the output level to be adjusted in software.

The board is fitted with a number of switches and encoders for control purposes. The board also has a miniature monochrome graphical display and a number of programmable LEDs including 2 x 8 LED VU meter LEDs.

The board is powered from 3.3V from USB or from a plug top power supply.

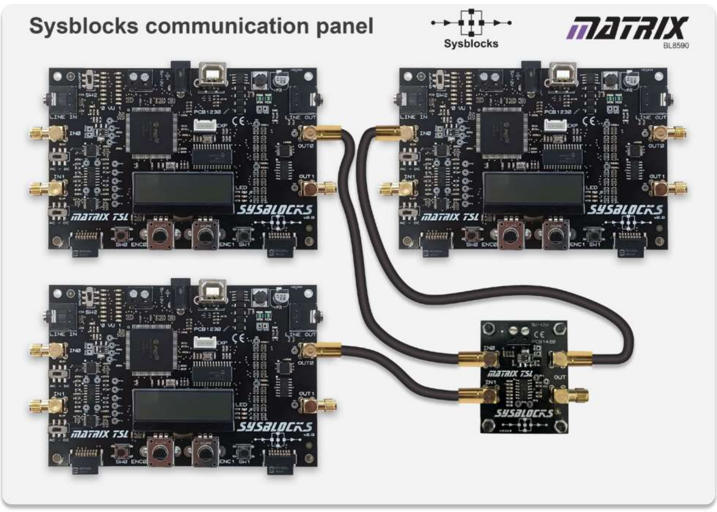
- Vin max 2.2V pk to pk
- Vout max 2.2V pk to pk

Hardware: Sysblocks products



BL8386 is our standard Sysblocks experimentation panel. It consists of a single Sysblocks board on a plastic panel with power supply and USB lead and is shipped in a standard tray for storage. The Sysblocks board is protected with a clear acrylic cover to make it more rugged in the lab.

Students use conventional oscilloscopes and spectrum analysers to examine the results of signal processing in a music and DSP context.



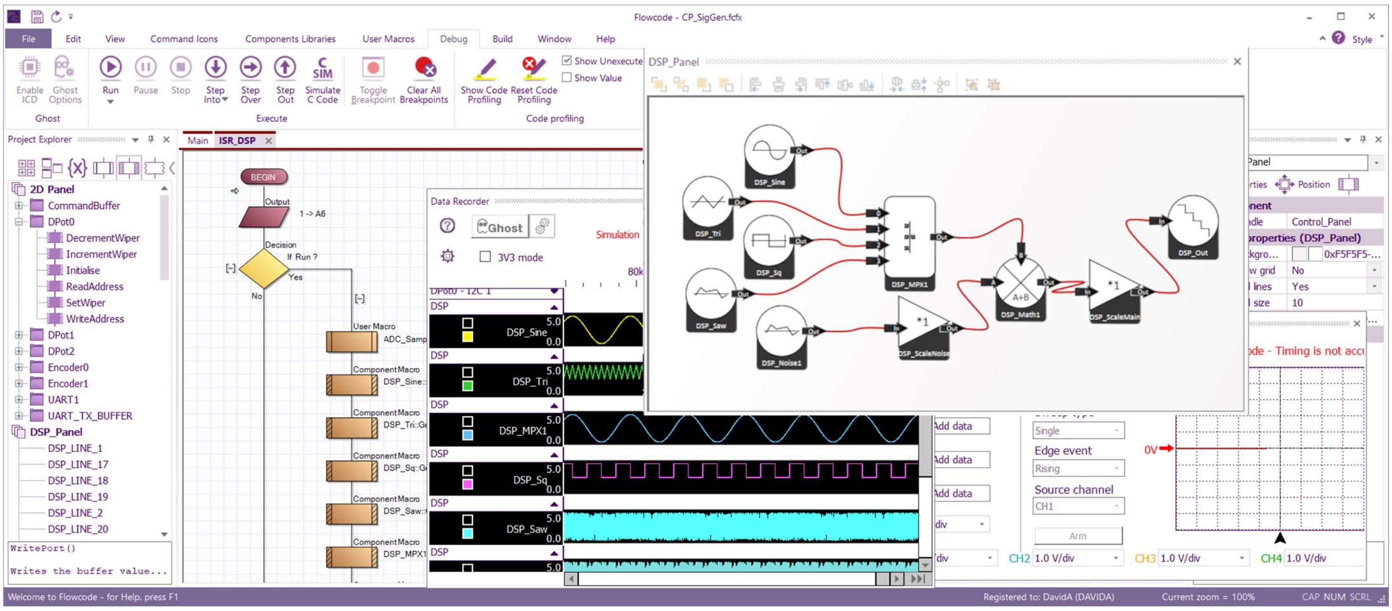
BL9296 is our Sysblocks Communications panel. This includes 3 x Sysblocks boards and a signal mixer board on a plastic panel. This configuration allows students to experiment with lots of different types of communications and examine their characteristics.

The first Sysblocsk board is used to modulate and / or encode a signal into a digital format. The second Sysblocks board is used to generate noise. The encoded/modulated signal and noise are mixed together in the mixer board and fed into the third Sysblocks board. Where the signal is then demodulated/decoded and fed to the outputs. Separate programs are used in each Sysblock. This configuration allows students to experiment with multiple modulating/demodulating and encoding/ deconding systems and examine their noise characteristics.

The Sysbocks Communications panel is supplied with a set of 6 micro BNC leads, USB lead and power supply in a standard tray. Sysblocks boards are protected with a clear acrylic covers.

Students use conventional oscilloscopes and spectrum analysers to examine the signals in the system.

Software: Flowcode



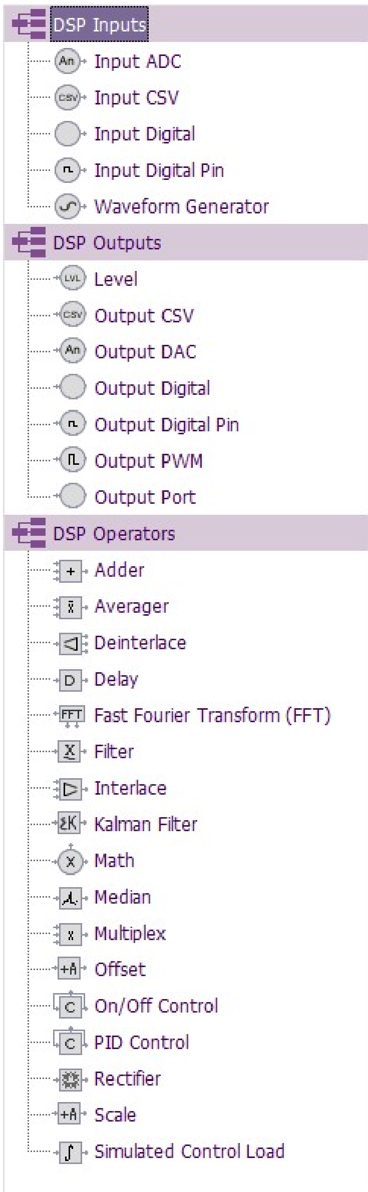
Flowcode 10 Electronic system design software includes Data flow / DSP components which you can see on the right. This programming paradigm allows students to build a wide variety of signal processing systems using conventional graphics that are used in text books and papers to describe the behaviour of signal processing systems.

To build a system students simply select the DSP / Data flow operator they want from the component library and drag it onto the panel. They then select the component's properties and connect the relevant inputs to the outputs of other icons to create a functioning system. They then connect inputs and outputs to the real world – A/D and D/A. Students can combine these data flow programs with flow charts, C code, pseudocode and state machine diagrams to produce highly functional electronic systems based on microcontrollers and Windows computers.

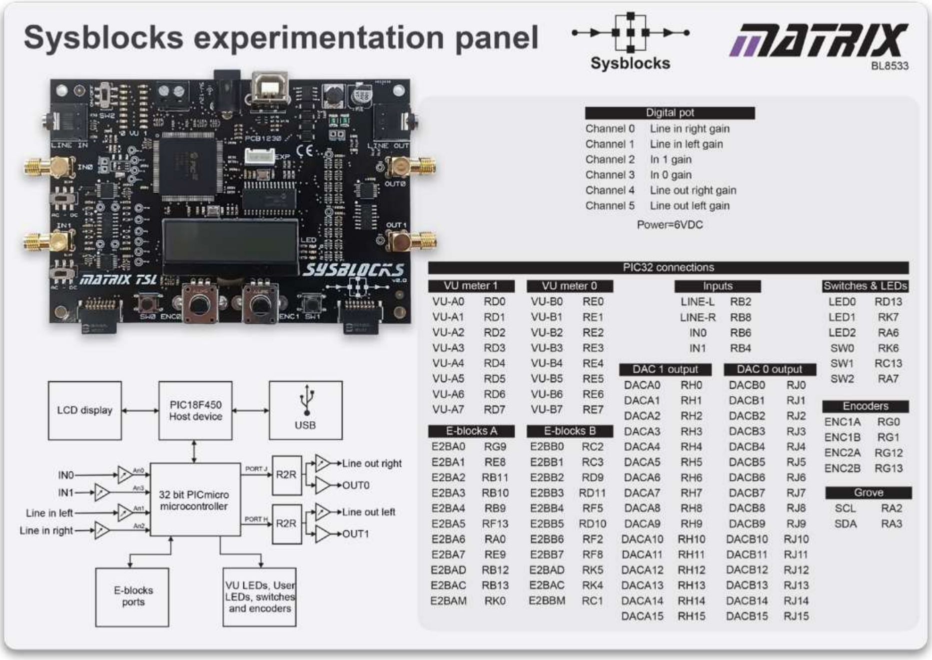
Once the system is constructed students can use the internal simulation engine to see the results of the program using the internal logic analyser and oscilloscope tools.

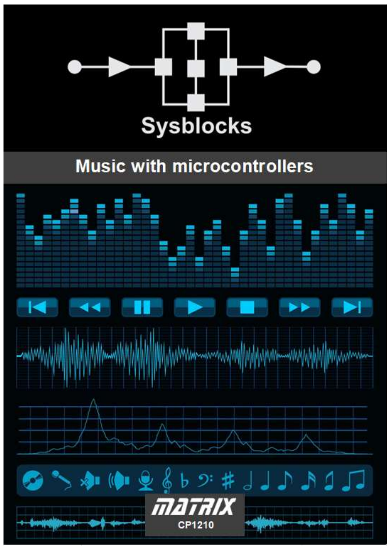
Once they are happy with their program they can compile it to the powerful Sysblock 32 bit PIC in just a few seconds. They can then use conventional oscilloscopes, spectrum analysers and other instruments to see their program working.

The images that follow demonstrate the kind of systems that can be developed with Flowcode 10.



Curriculum: Music with microcontrollers course





Description

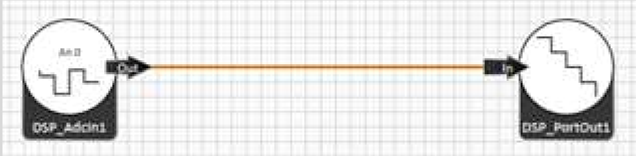
The Music Technology pack uses a single Sysblock and Flowcode Embedded to allow students to quickly construct a wide variety of digital music applications and understand the principles of digital music manipulation and effects. Students will gain an appreciation of D/A conversion and A/D conversion and Digital Signal Processing Techniques that allow music to be sampled, delayed, manipulated and produces in the digital domain. Students make use of signal generators, microphones and speakers/headphones to create and understand a variety of applications in Music Technology. The pack is suitable for students 16+ with an interest in embedded programming and Music.

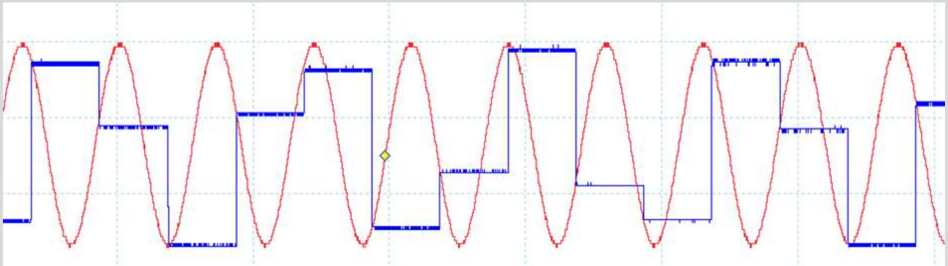
Learning objectives

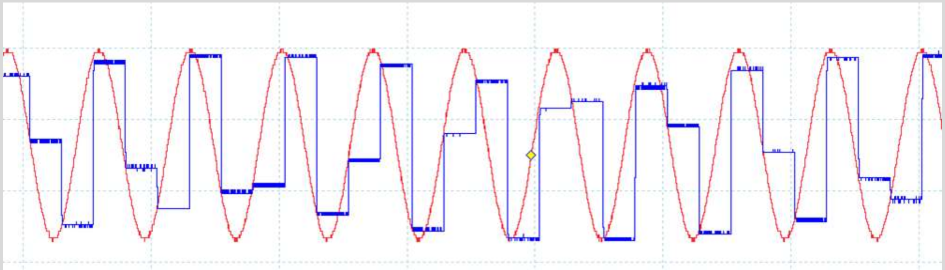
- Analogue to Digital Conversion
- Digital to Analogue conversion
- Simple Digital Signal Processing
- Nyquist
- Audio effects including echo, reverb, distortion
- Mixers
- Sampling and music creation
- Delays and stadium balancing
- EQ and filters

Example: Nyquist

Simply connecting the input to the output allows students to study sampling techniques in digital audio systems. Varying the sample rate allows students to understand the Nyquist criteria. Shown here are the simple data flow program and oscillograms for Nyquist compatible and incompatible sample rates.

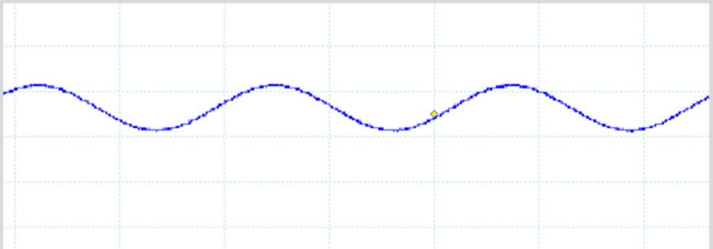


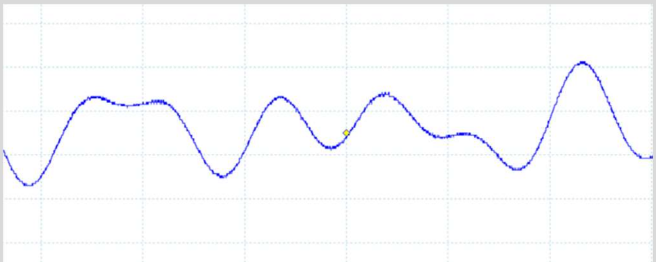


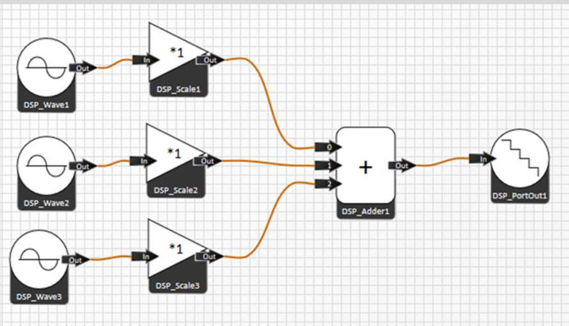


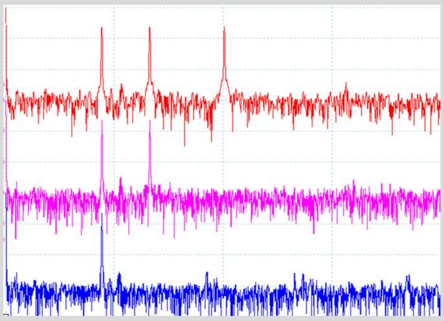
Example: Audio mixer

Students can understand how digital audio signals are constructed and manipulated. In this case a mixer program allows students to mix together different frequencies and waveforms to create musical chords and study their shape on an oscilloscope and their spectral plot. Shown here are the waveforms for note A and chord A with spectral plot.

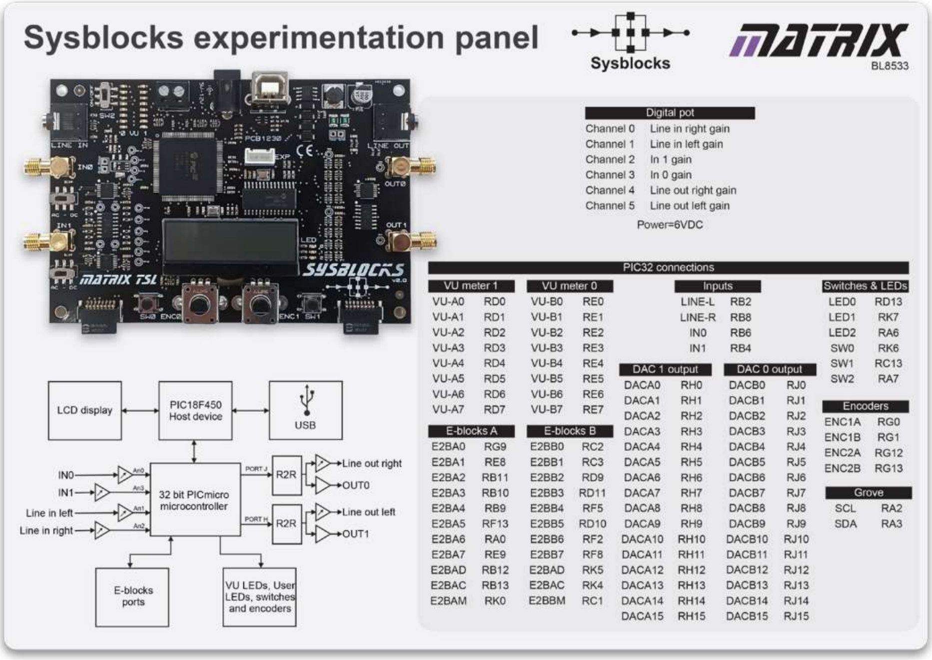


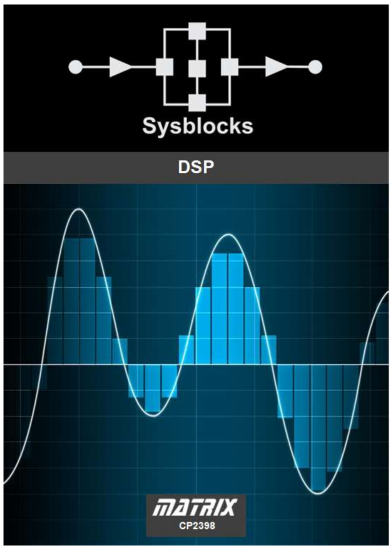






Curriculum: Systems, signals, DSP, FFT





Description

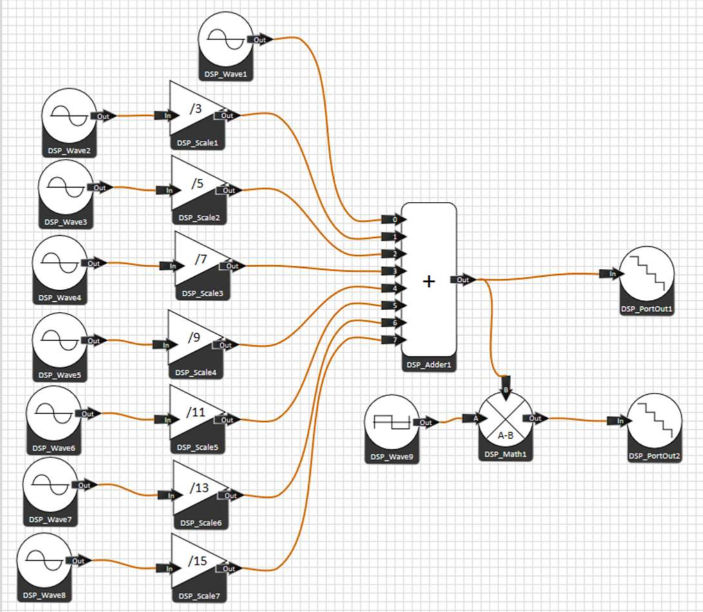
The Systems, signals, DSP and FFT pack introduces students to a number concepts in digital systems including A/D and D/A conversion, Nyquist, Digital Signal Processing, signal manipulation, Fourier analysis of signals and the FFT algorithm, and Digital Filters. Students perform a range of experiments that teach them about the details of signal manipulation using a single Sysblock, pre- written programs created in Flowcode embedded, a signal generator, an oscilloscope and a spectrum analyser. Students download the programs and carry out a number of experiments that teach them about the parameters and techniques of Signal processing. There is also plenty of chance for creative programming using Flowcode embedded for the advanced students who will find that the Sysblocks and Flowcode combination provide the perfect platform for the rapid creation of signal manipulation systems.”

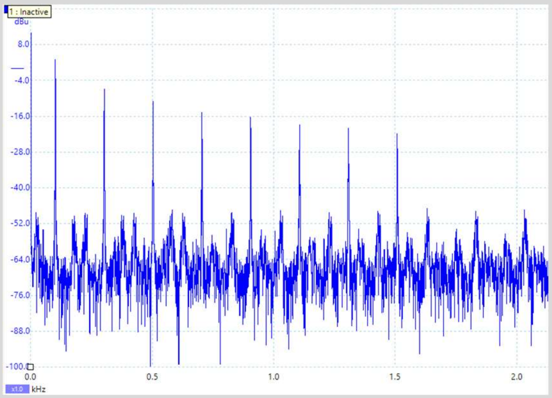
Learning objectives

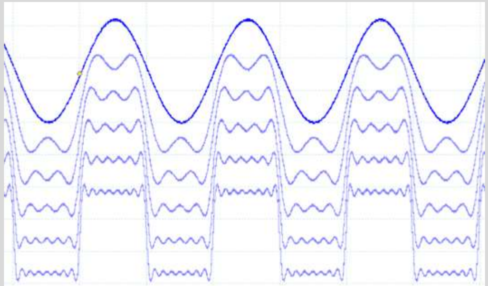
- Analogue to Digital Conversion
- Digital to Analogue conversion
- Nyquist
- Digital Signal Processing techniques
- Digital Signal manipulation
- Convolution and level detection
- Fourier analysis
- FFT systems
- Digital filters

Example: Theory of Fourier

This example shows how harmonics can be added to create a square wave. Images show the data flow diagram, a composite oscillogram with different numbers of harmonics and a typical spectral plot.

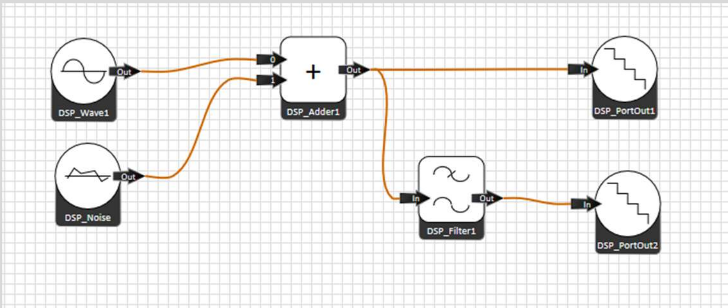


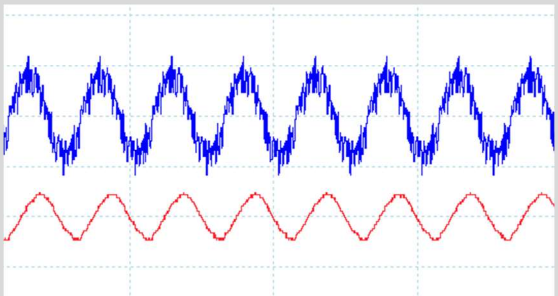


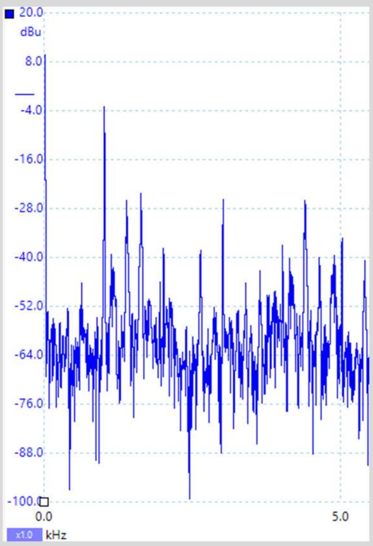


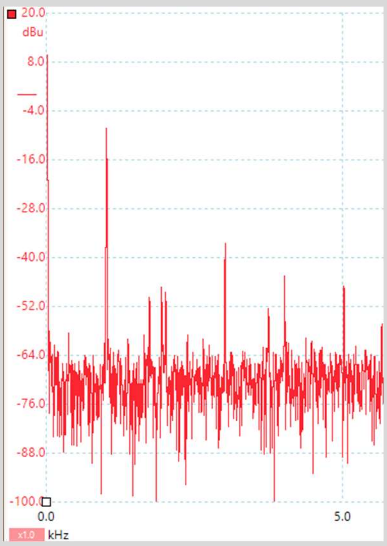
Example: Digital filters

Students can construct a wide range of different types of digital filter - including FIR and IIR types - and study their performance. The images show the simple data flow program, and oscillograms and spectral plots of signals before and after filtering.

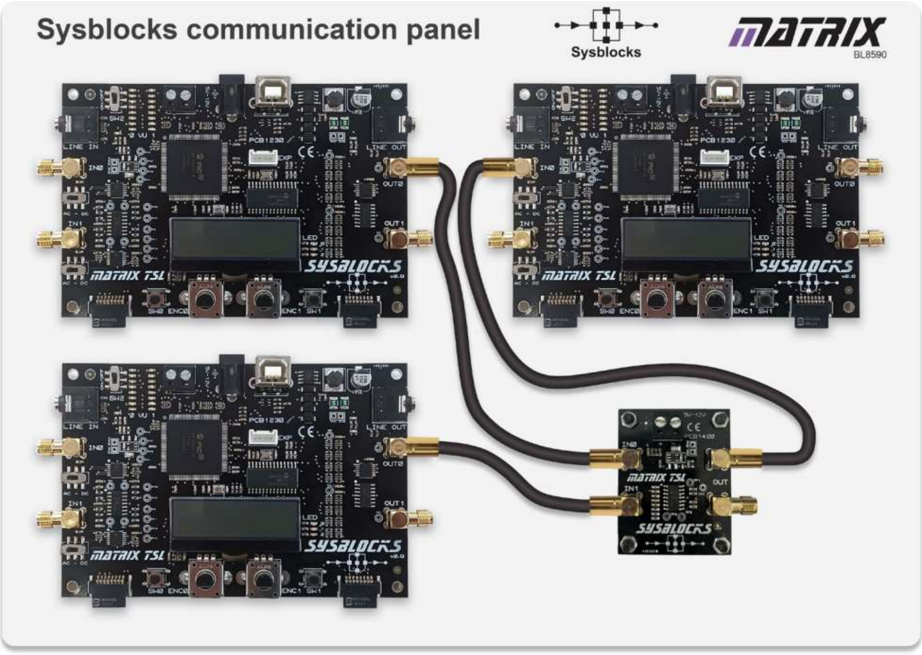


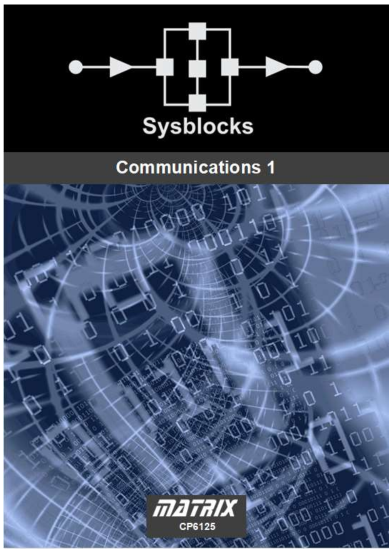






Curriculum: Communications and digital radio techniques



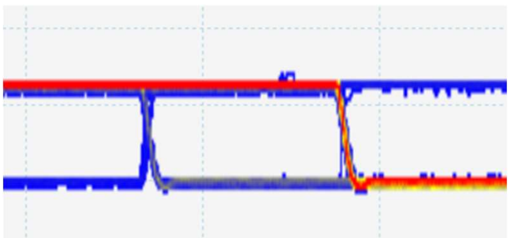


Description

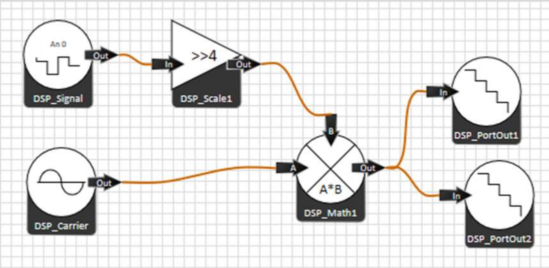
Once students have been through the Systems, signals, DSP and FFT pack they can move onto the Communications and digital radio techniques course. This course allows students to construct and experiment on a wide range of systems based on Sysblocks. Students use a panel with three Sysblocks on to create a sequence of communications modulators and encoders, add noise, and then demodulate / decode the resulting signals. In doing so students understand the parameters of communications systems including Phase Locked Loop performance, signal to noise ratios, bit error rates and the principles of software defined radio. The communications systems students investigate include PLLs, AM, FM, OOK, ASK, FSK, BPSK, QAM, QPSK and DSSS. Students can also examine spread spectrum systems and correlation. Students work with pre-written programs for Sysblocks. There is also plenty of chance for creative programming using Flowcode for the advanced students who will find that the Sysblocks panel and Flowcode combination provide the perfect platform for experimentation with digital comms. The hardware platform includes 3 x Sysblocks panels (encode/ modulate, decode/demodulate, and noise generator) and a mixer board.

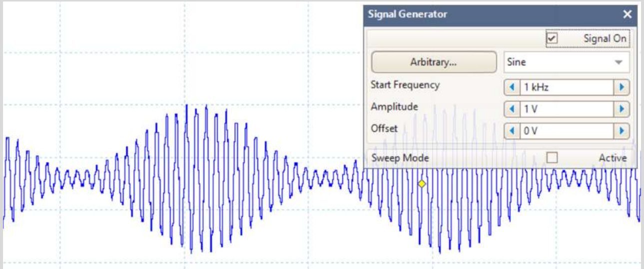
Learning objectives

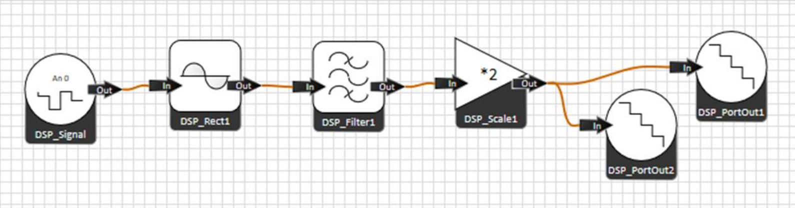
- AM and FM
- Digital communications systems
- Modulation / Demodulation
- Encoders / Decoders
- Phase Locked Loops
- PRBS and Bit Error Rates
- AM, FM, OOK, ASK, FSK, BPSK, QAM, QPSK and DSSS
- Digital Radio Techniques
- Spread spectrum techniques



Eye diagrams can be produced for OOK, ASK, FSK and BPSK.

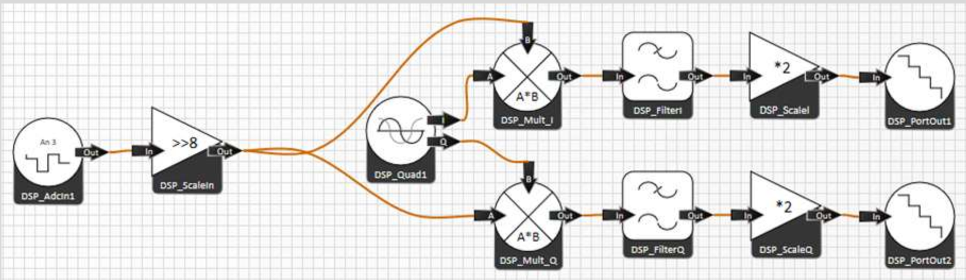






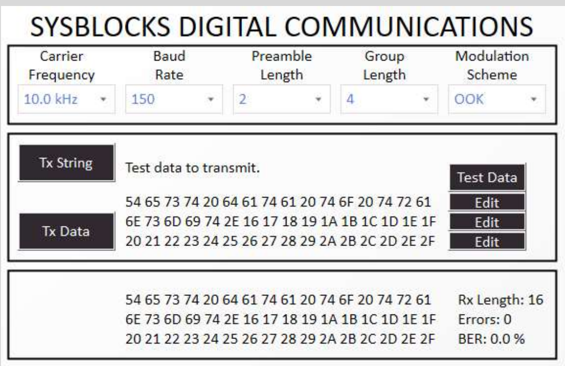
Example: AM transmitter receiver

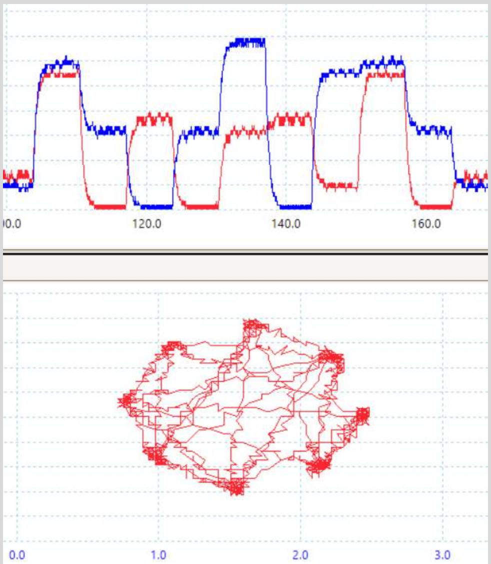
The equipment allows students to set up AM and FM systems that show some of the principles of digital radio technology. Here you can see the AM transmitter and receiver Flowcode data flow programs, an oscillogram of a 1kHz modulated wave and the associated spectral plot.



Example: 8-PSK detector

With the equipment it is possible to set up various kinds of modulator encoder - demodulator -decoder systems: in this case 8PSK. The first image here shows the Data Flow program in Flowcode. Accompanying code sets up data packets. Also shown here are the I and Q outputs, the constellation plot, and the Bit Error Rate counter program.

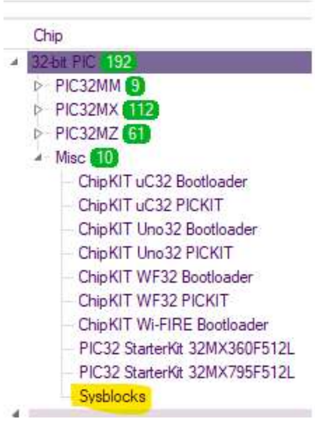




Your first Sysblocks program

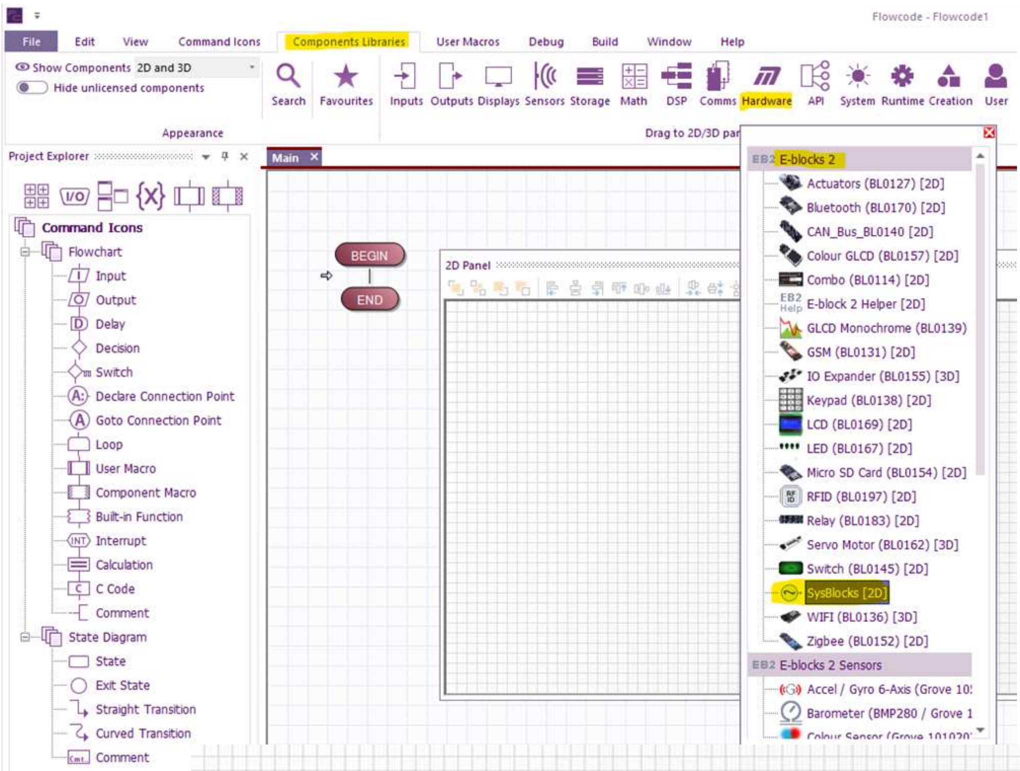
1 Create a new project

Choose a target for a new project. You can find the SysBlocks by the path **32-bit PIC->Misc->SysBlocks.**

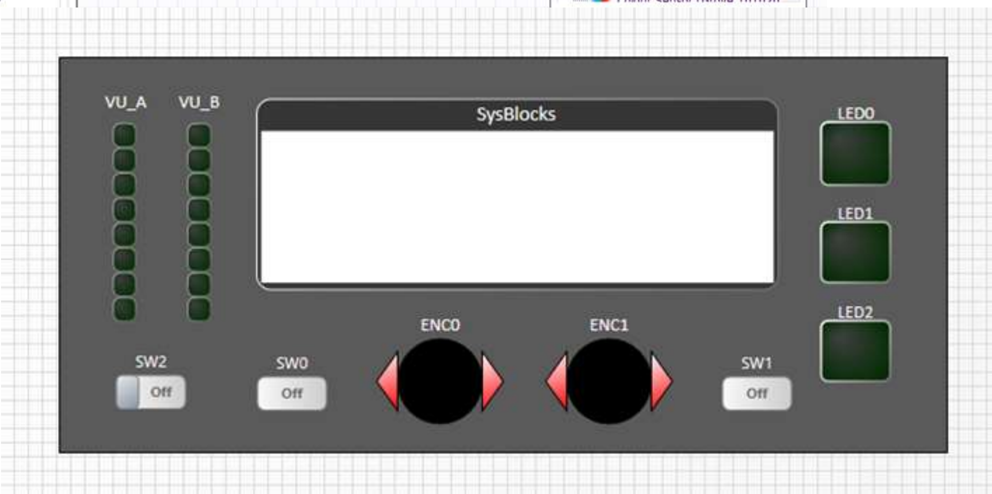


2 Add the Sysblocks control panel to the 2D panel

One of the potential difficulties with the Sysblocks architecture is that there are two processors on the board that handle different tasks. Flowcode takes care of this through a hardware component that allows access to the switches, display, and encoders. A single component allows easy access to all of them for the user.



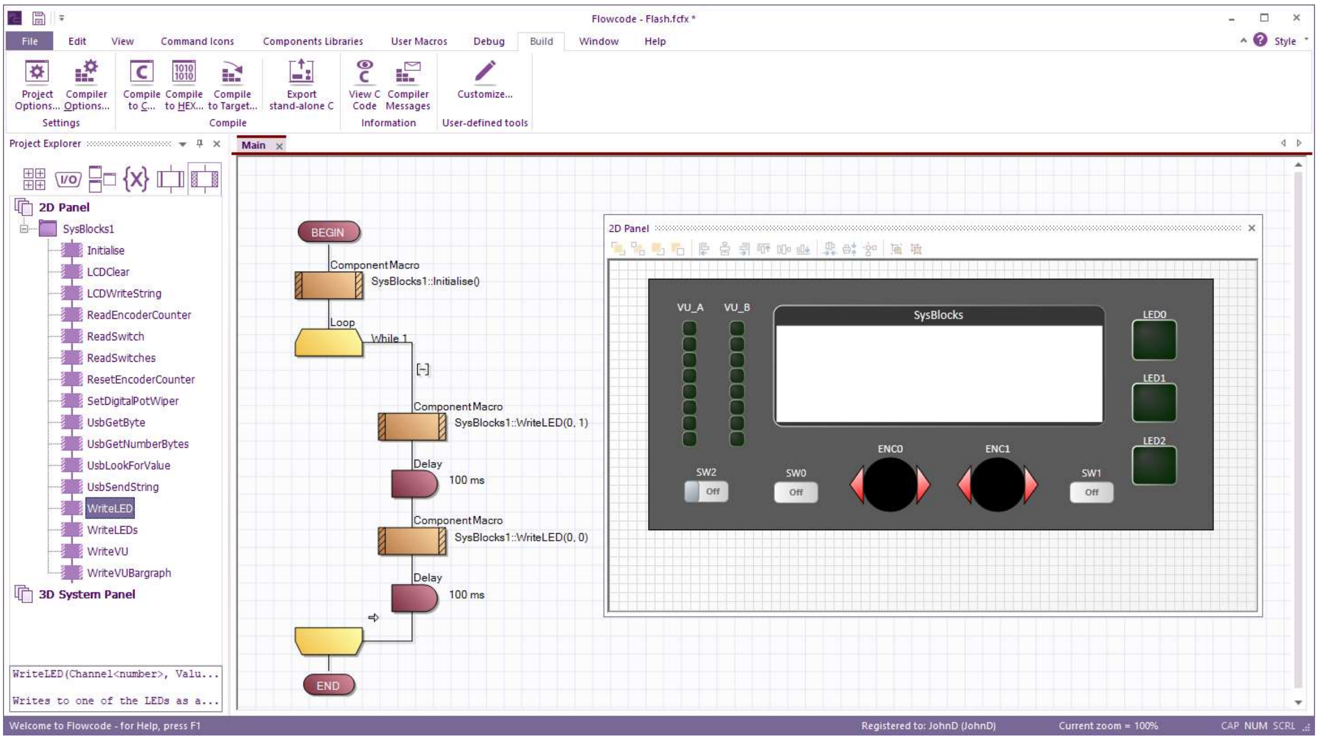
From Component Libraries->Hardware->E2Blocks->SysBlocks.



Your first Sysblocks program

3 Develop your program

Use a component macro to initialise the Sysblocks component. Create a loop - while 1 - and Use Sysblocks LED component macro, and a couple of 100ms delays, to flash LED 1 on and off.

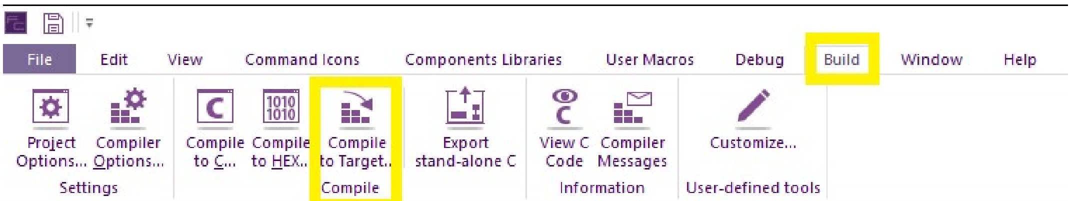


4 Simulate your program

Select DEBUG...RUN to simulate the program

5 Program the Sysblocks board

Select BUILD...COMPILE TO TARGET to send your program to the Sysblocks board.

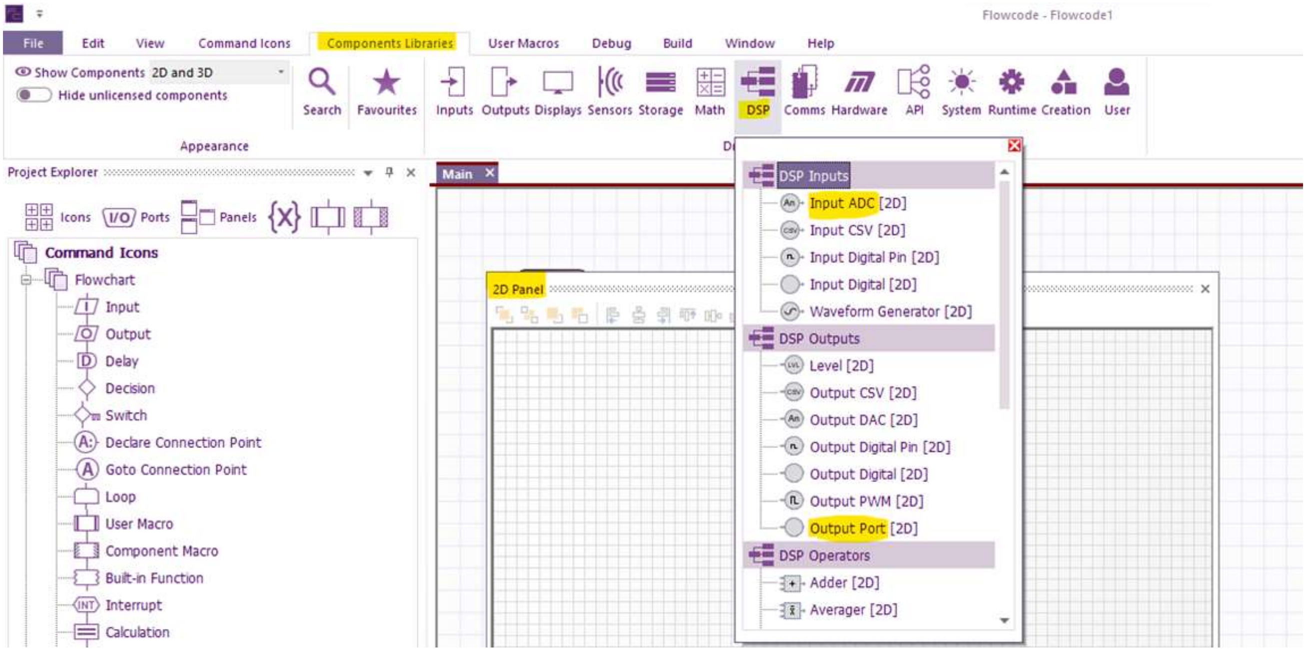


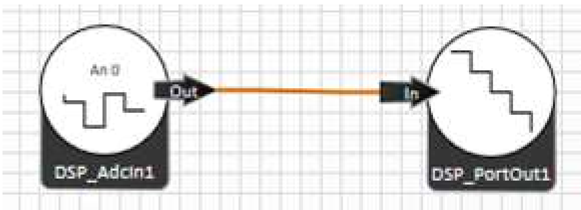
Congratulations! You have built your first Sysblocks program. LED 0 should be flashing at 1 second intervals.

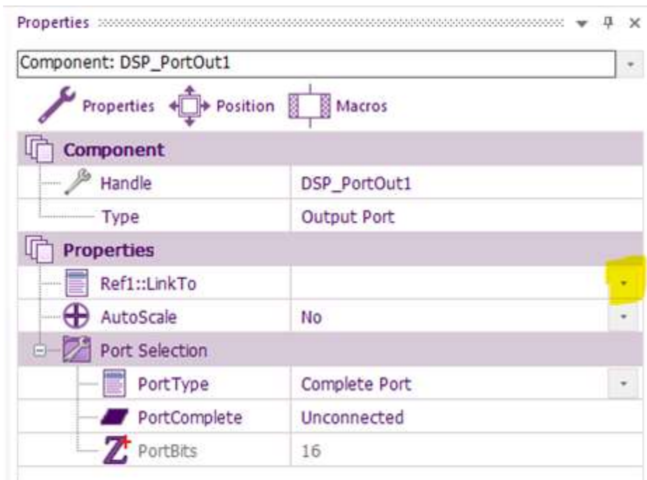
Your first Sysblocks program

6 Create a Data flow program

For a Data flow program you need to add at least one **Input ADC** and one **Output Port** from the Components Libraries. You can find the **Input ADC** and **Output Port** in the **DSP** section of Flowcode which is under the **Components Libraries**. You need to drag them to the **2D Panel**.







After dragging them to the 2D Panel, you need to connect the components **DSP_PortOut** and **DSP_AdcIn** together by setting up the properties of **DSP_PortOut.** The properties would be shown in the right if you click the component **DSP_PortOut** in **2D Panel**. Click the triangle of **the Ref1::LinkTo** to connect the component **DSP_PortOut** to the component **DSP_AdcIn**. If they are connected, you will see a line between them.

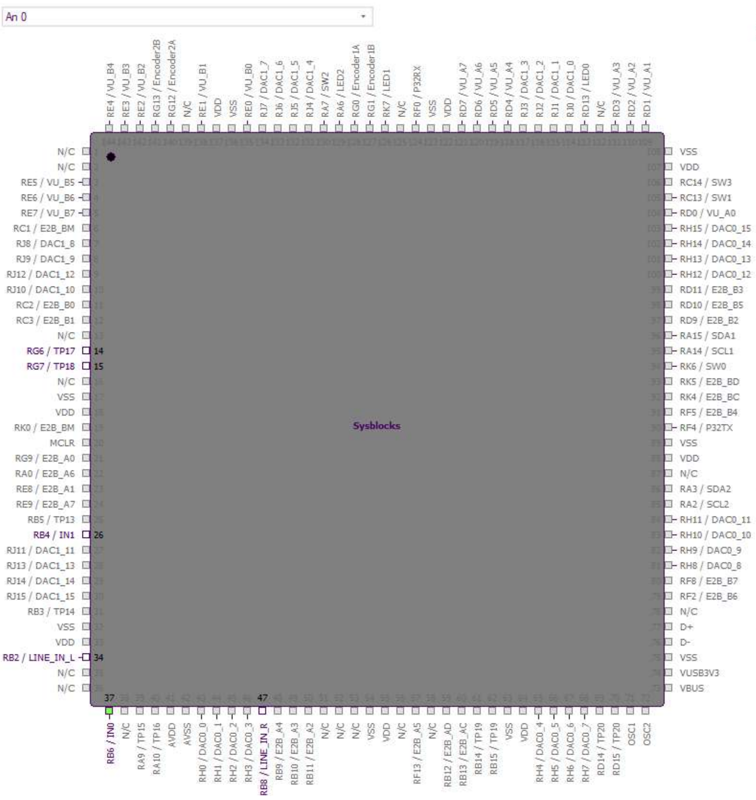
Select Autoscale for DSP_PortOut to YES. The sampling is 12 bit but the output is 16 bit and Autosclale takes care of that.

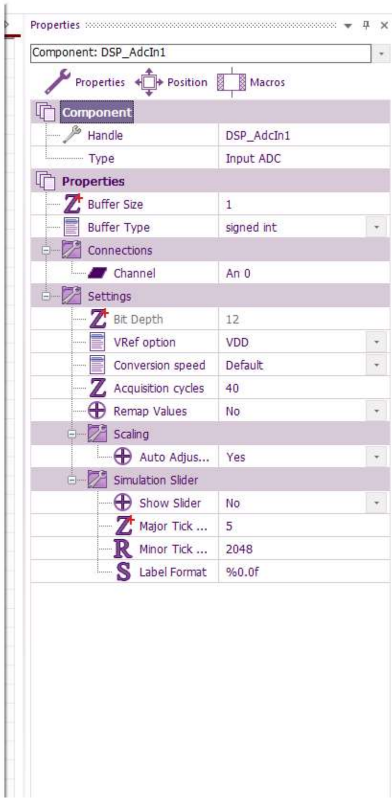
Of course this is a very simple program - we take an input, sample it at 3kHz to create a digital data stream, we pass that to an output DAC and recreate the signal. Its not a very useful program, but it gets the system up and running and you can then add other data processing icons to develop a more complex system.

Your first Sysblocks program

7 Connect the input port

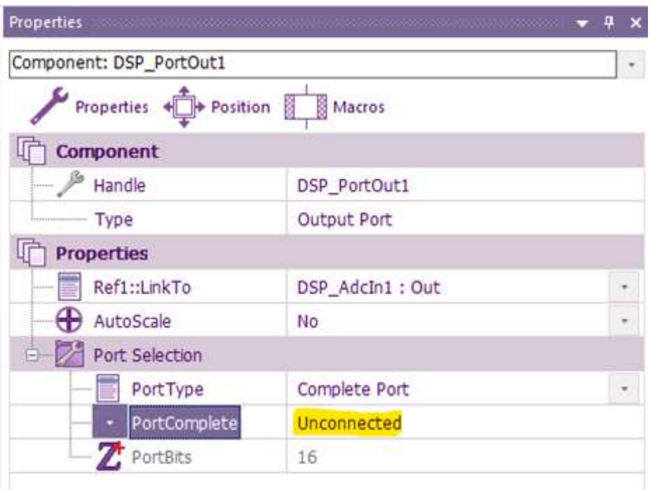
You need to set up the input in the hardware of the Sysblocks board by right clicking the **DSP_AdcIn** in 2D and selecting Properties. Under CONNECTIONS … CHANNEL select the input port you want. We have selected AN0 which is the Micro BNC IN0.

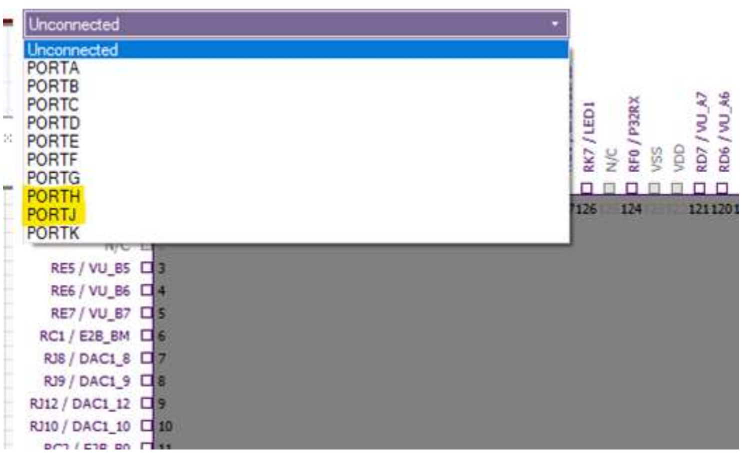




8 Connect the output port

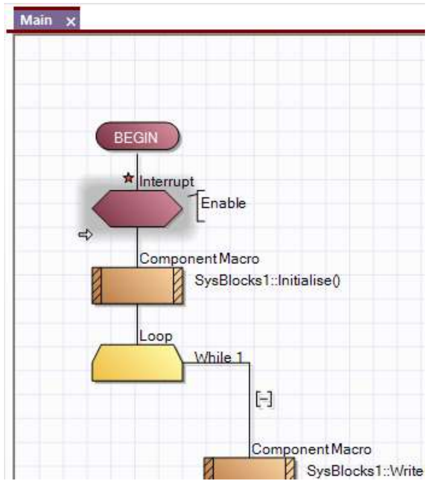
You need to set up the mapped Port in the hardware of the Sysblocks by click the **PortSelection** in the properties of DSP_PortOut. The two output ports of the Sysblocks board could be Port J (OUT0) and Port H (OUT1) , which you can check with the **User Drawing** at the Appendix of this document. Just choose one mapped port for the Output. Also, you need to set the **AutoScale** as **Yes** in the Properties of **DSP_PortOut** to make sure the output data does not overflow.



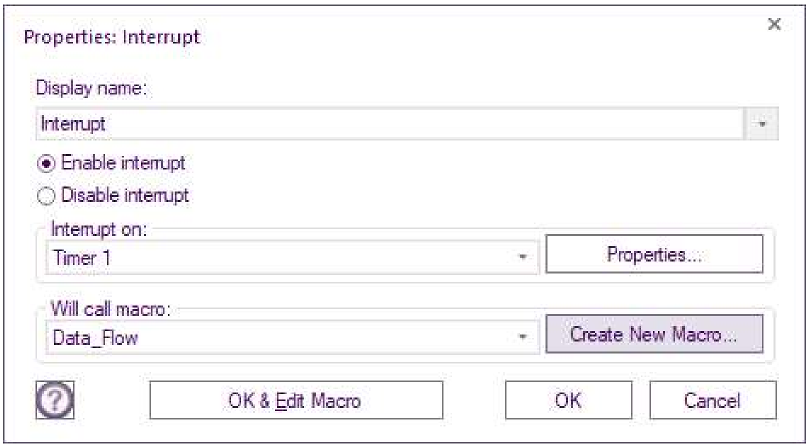


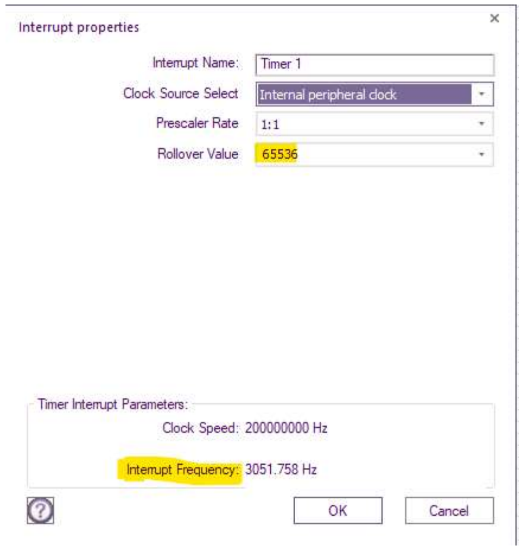
Your first Sysblocks program

9 Setting up the interrupt



Add an Interrupt icon to the start of your program. Double click to set the properties. Select **Timer 1 to call a new macro: Data-flow**. Click on CREATE NEW MACRO to create the new Data-flow macro.





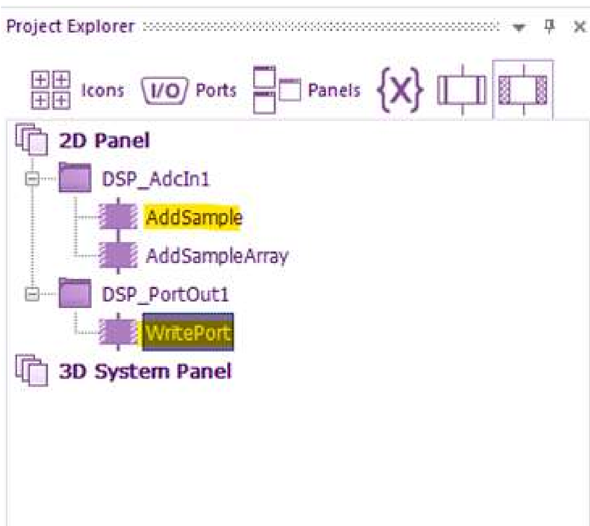
Set the Properties of the interrupt so that the interrupt frequency is 3051Hz.

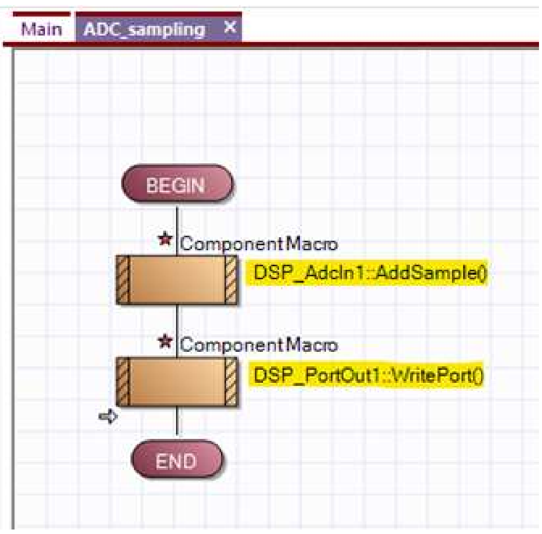
This is now the ‘Tick’ frequency for your Data flow program.

10 Your Data flow program

Add the **AddSample** and **WritePort** to the Data_flow macro. Every time there is an interrupt these routines in the Data flow program will be executed. You can now send the program to the board as in step 5.

Note that you still have your Main flow chart program executing: so LED0 is flashing at regular intervals at the same time as the Data Flow program is running.





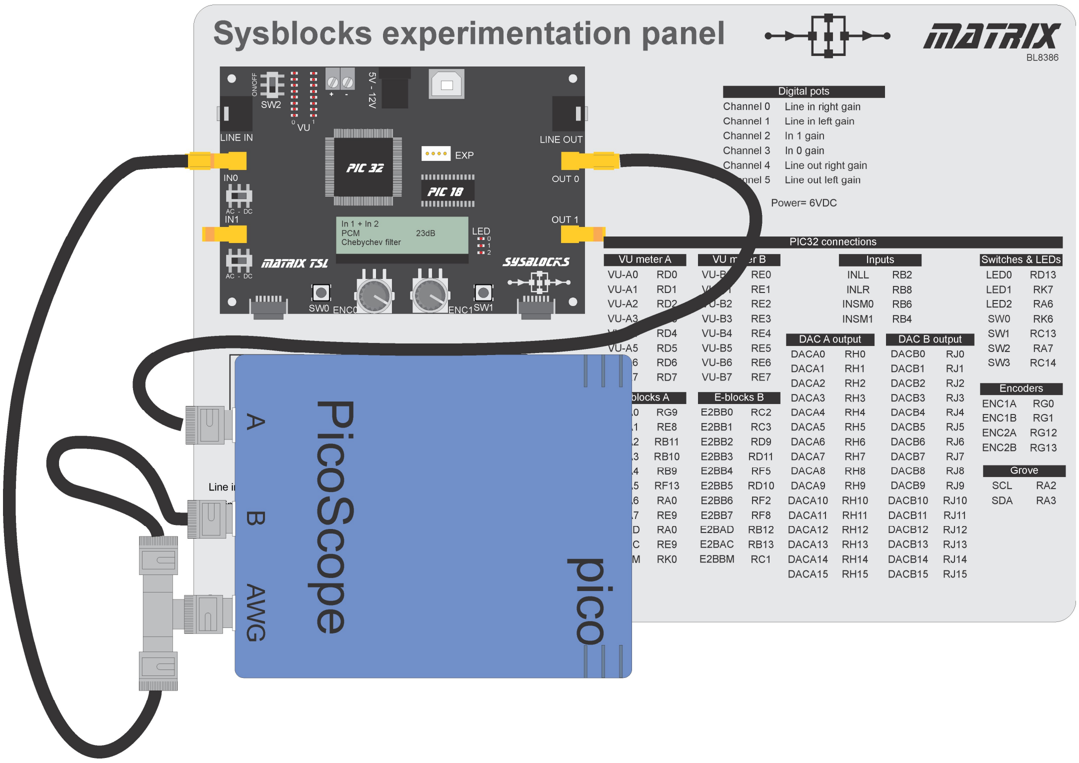
Your first Sysblocks program

11 Evaluating the first program

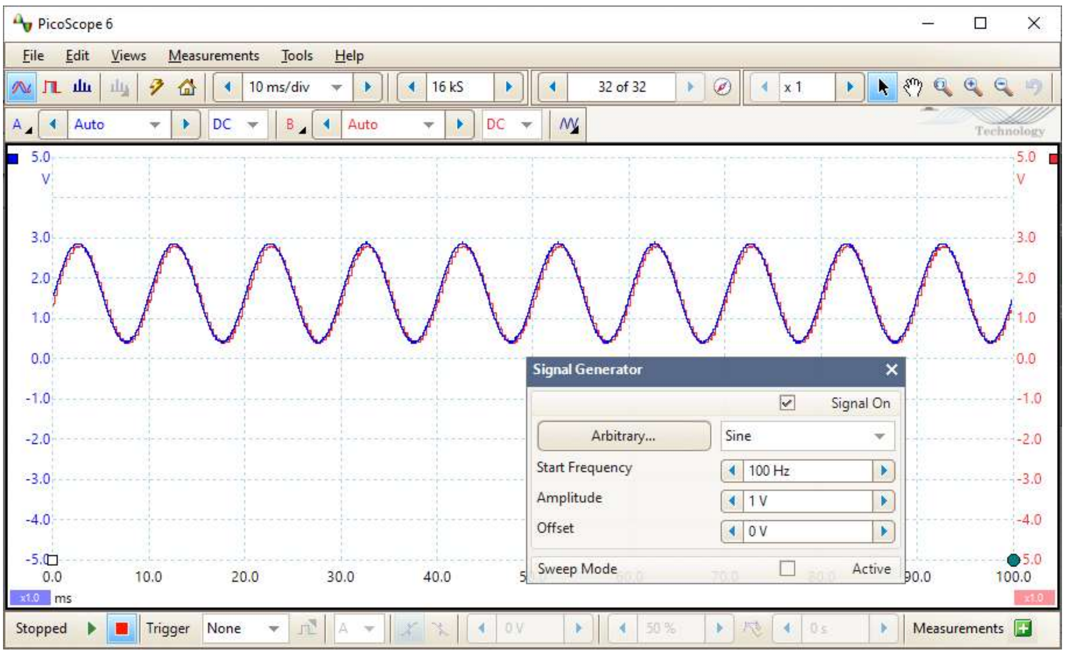
For this section you will need a signal source and a two channel oscilloscope like a Picoscope.

The Pico scope has a waveform generator that you can use as a signal source. You will need a splitter so that you can feed this into channel A of the scope and IN0 on the Sysblocks board. Set switch SW4 to AC coupling.

You can connect OUT0 to channel B of the oscilloscope.



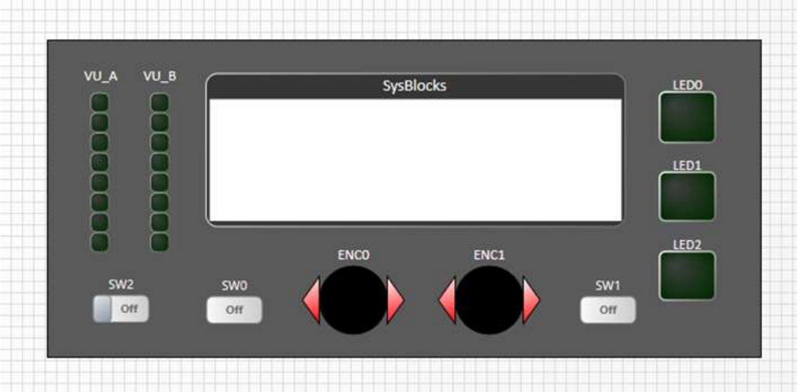
Set the waveform generator to produce a sinewave at a frequency of 100Hz. Your Data flow program will sample this at 3kHz and will send the sampled waveform to the output. You should see this on your oscilloscope:



What you can see here is that the Sysblocks board has sampled the input (blue) and reflected it on the output (red).

Now that you have the basic system up and running you are ready to start to manipulate signals.

The Sysblocks Flowcode component



Sysblocks has two processors - the main 32 bit PIC processor that executes the program and a secondary 18 series PIC that takes care of USB communications and the LCD.

This architecture allows the 32bit PIC freedom to focus on number crunching and not I/O tasks.

The potential downside of this architecture is that using the LCD could be complex. The complexity is taken care of for you by the Sysblocks component. This includes all of the hardware routines that you need to control the board.

Initialise ()

Must be called at the start of any Sysblocks program to initialise the hardware functions

LCDClear()

Clears the LCD display

LCDWriteString()

Allows you to write a message to the LCD. Max 20 characters per row.

X BYTE Starting column for the text.

Y BYTE Row to write text on (0..3)

Data <-STRING The string to write.

ReadEncoderCounter()

Read one of the two encoders on the board.

Channel BYTE Encoder to read. 0=ENC0, 1=ENC1

RETURN INT The current value of the encoder

ReadSwitch()

Read one of the three switches on the board.

Channel BYTE Switch to read. 0=SW0, 1=SW1, 2=SW2

RETURN BOOL The state of the switch.

ReadSwitches()

Read all three switches at once.

RETURN BYTE States of all switches. Bit0=SW0, Bit1=SW1, Bit2=SW2

ResetEncoderCounter()

Make the current one of the encoder counters zero.

Channel BYTE Encoder to reset. 0=ENC0, 1=ENC1

SetDigitalPotWiper()

Set the position of one of the digital potentiometers on the board. When the board powers up, all the potentiometers are set to the mid-way position. For most applications, it is not necessary to alter them.

Channel BYTE The channel to set: -

0 Line in left gain.

1 Line in right gain

The Sysblocks Flowcode component

2 IN 0 gain

3 IN 1 gain

4 Line out left level

5 Line out right level

Value	UINT	The wiper setting from 0 to 128.

USBGetByte()

Returns the next available byte from the USB virtual com port. This provides a way for an application running on a PC to send data to the SysBlocks firmware.

RETURN	UINT	The byte read from the stream.

USBGetNumberBytes()

Returns the number of bytes available in the stream from the USB host.

RETURN	UINT	Number of bytes available.

USBLookForValue()

Peek into the stream from the USB host to look for a specific value or sequence of values. This can be used to respond to commands that are marked by delimitiers.

Value	<- STRING	String containing the character or sequence to search for.
NumChars	BYTE	The number of characters in the sequence.
RemoveContent	BYTE	0 = Just search, leave the stream unaltered.
		1 = Remove bytes up to and including the search term.
ResetFind	BYTE	0 = If removing bytes, then remember if the first part of the
		sequence has been found.
		1 = Ignore partial sequences and start again.
RETURN	BYTE	0 = Not found
		1 = Found

USBSendString()

Send a string to the USB host.

Data	STRING	The string to send.

WriteLEDs

Set the states of all three of the LEDs that are to the right of the LCD.

Values	BYTE	A binary number where the first three bits represent the states of the three LEDs.

WriteVU()

Set the state of one of the banks of eight LEDs at the upper left corner of the board.

Channel	BYTE	The channel, 0 or 1.
Value	BYTE	Eight-bit binary number to be shown by the LEDs. Bit 0 is the lowest LED and bit 7 is the topmost.

WriteVUBargraph()

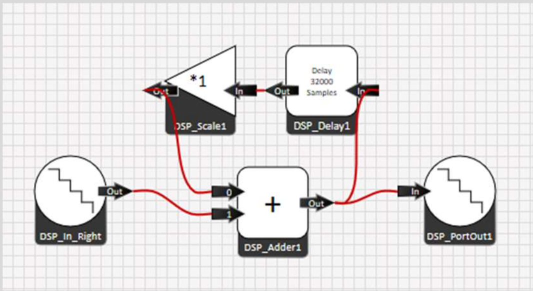
Channel	BYTE	The channel, 0 or 1.
Value	BYTE	The number of LEDs to be lit.

DSP icons

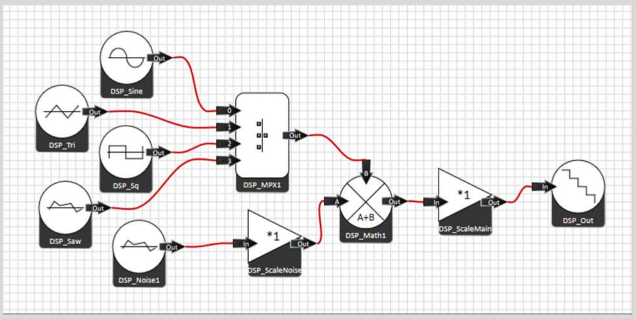
At the time of going to press the image on the right shows the DSP icons currently available.

These are documented in the Flowcode wiki which can be found at www.flowcode.co.uk/wiki.

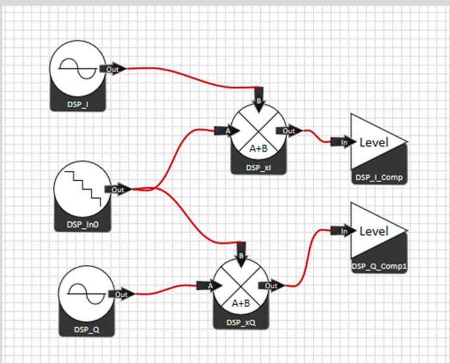
You can use these icons by dragging them onto the 2D panel and connecting them together. Connections are made between the icons using the properties panel of each icon. When connected you will see an appropriate Data flow line between icons.



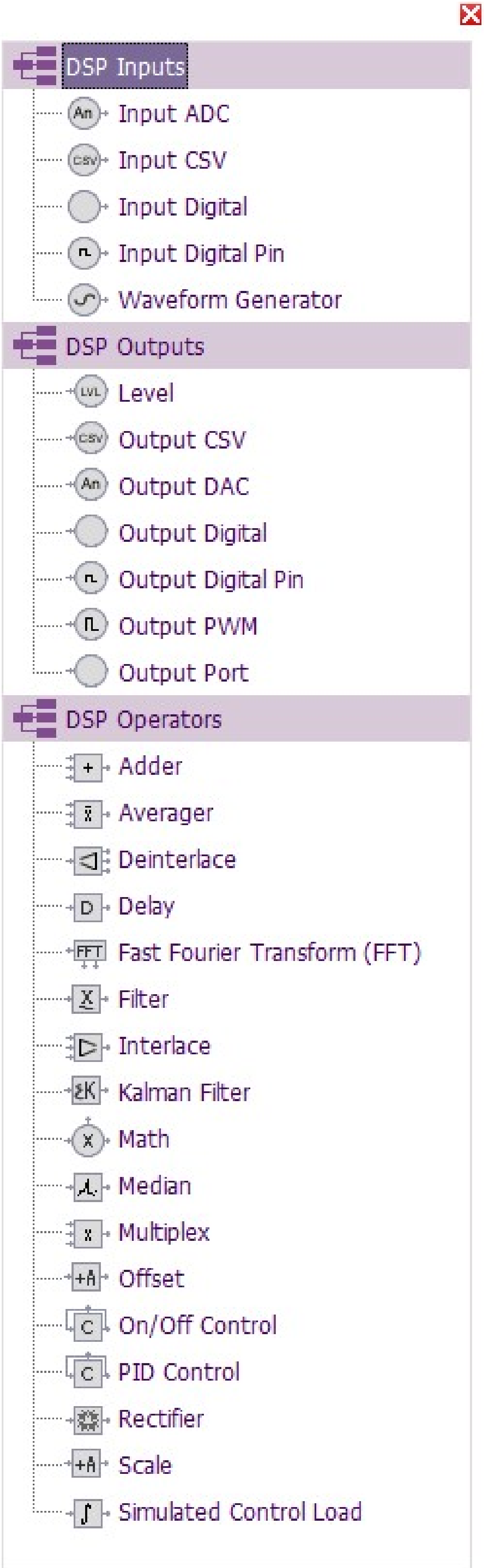
Echo



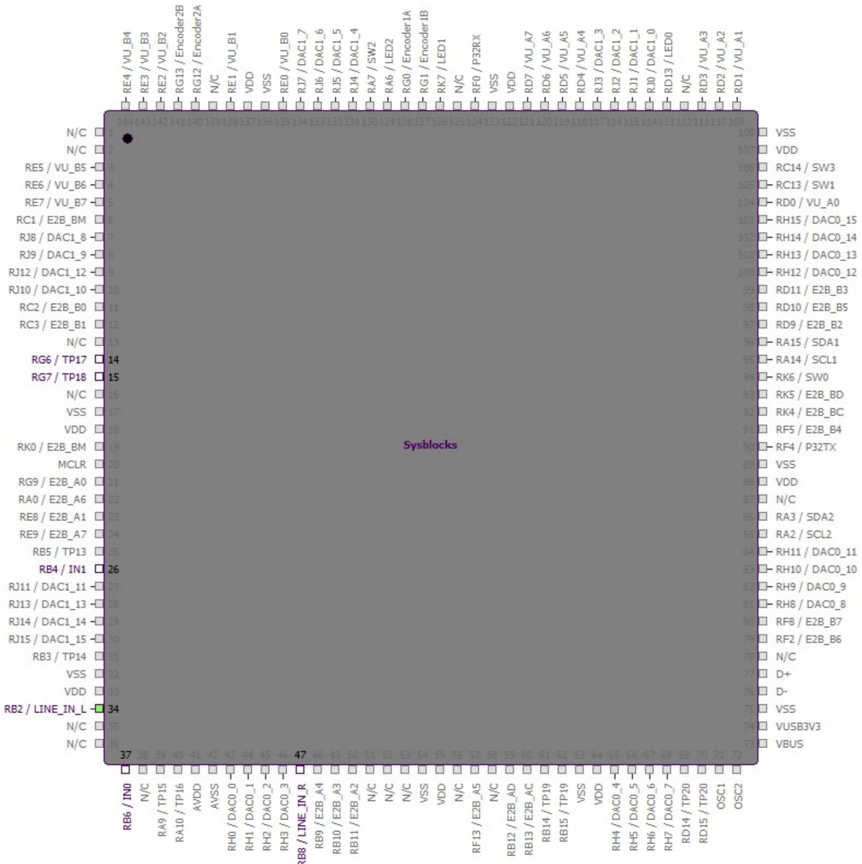
Signal generator



QPSK modulation



Connection tables



PIC32 connections

VU meter A

VU-A0	RD0
VU-A1	RD1
VU-A2	RD2
VU-A3	RD3
VU-A4	RD4
VU-A5	RD5
VU-A6	RD6
VU-A7	RD7

VU meter B

VU-B0	RE0
VU-B1	RE1
VU-B2	RE2
VU-B3	RE3
VU-B4	RE4
VU-B5	RE5
VU-B6	RE6
VU-B7	RE7

Inputs

LINE-L	RB2
LINE-R	RB8
IN0	RB6
IN1	RB4

Switches & LEDs

LED0	RD13
LED1	RK7
LED2	RA6
SW0	RK6
SW1	RC13
SW2	RA7
SW3	RC14

E-blocks A

E2BA0	RG9
E2BA1	RE8
E2BA2	RB11
E2BA3	RB10
E2BA4	RB9
E2BA5	RF13
E2BA6	RA0
E2BA7	RE9
E2BAD	RA0
E2BAC	RE9
E2BAM	RK0

E-blocks B

E2BB0	RC2
E2BB1	RC3
E2BB2	RD9
E2BB3	RD11
E2BB4	RF5
E2BB5	RD10
E2BB6	RF2
E2BB7	RF8
E2BAD	RB12
E2BAC	RB13
E2BBM	RC1

DAC A output

DACA0	RH0
DACA1	RH1
DACA2	RH2
DACA3	RH3
DACA4	RH4
DACA5	RH5
DACA6	RH6
DACA7	RH7
DACA8	RH8
DACA9	RH9
DACA10	RH10
DACA11	RH11
DACA12	RH12
DACA13	RH13
DACA14	RH14
DACA15	RH15

DAC B output

DACB0	RJ0
DACB1	RJ1
DACB2	RJ2
DACB3	RJ3
DACB4	RJ4
DACB5	RJ5
DACB6	RJ6
DACB7	RJ7
DACB8	RJ8
DACB9	RJ9
DACB10	RJ10
DACB11	RJ11
DACB12	RJ12
DACB13	RJ13
DACB14	RJ14
DACB15	RJ15

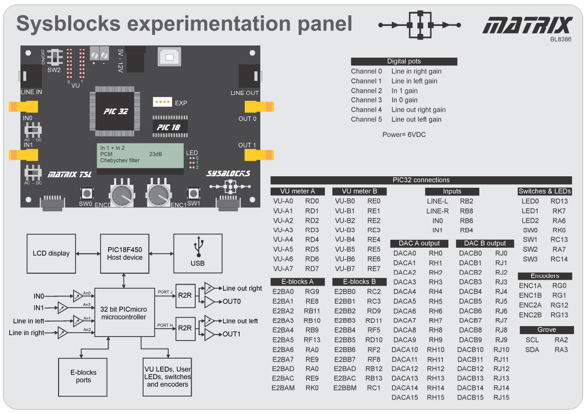
Encoders

ENC1A	RG0
ENC1B	RG1
ENC2A	RG12
ENC2B	RG13

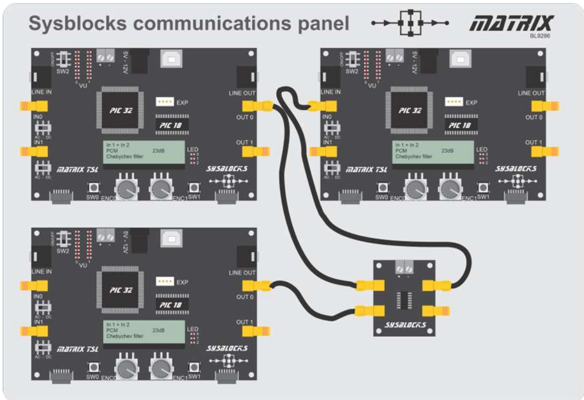
Grove

SCL	RA2
SDA	RA3

Product codes



BL8386 standard experimentation panel with power supply and USB lead in standard tray. Sysblocks board is protected with a clear acrylic cover. E-blocks II compatible.



BL9296 Communications panel including 3 x Sysblocks boards, mixer board, set of 6 micro BNC leads, USB lead and power supply in a standard tray. Sysblocks boards are protected with a clear acrylic covers. E-blocks II compatible.

BL2461 Sysblocks PCB only

BL9731 Sysblocks mixer PCB only

CP1210 Music with microcontrollers curriculum

CP2398 Systems, signals, DSP, FFT curriculum

CP6125 Communications and Software Defined Radio

BL8386 Standard Sysblocks experimentation system (Music with microcontrollers or DSP)

BL9296 Sysblocks Communications experimentation system

BL6374 BNC splitter

BL4585 BNC to SMA adaptor

BL6889 SMA to SMA lead 250mm

BL0167 E-blocks II LED board

BL0145 E-blocks II Switch board

BL0117 E-blocks II Prototype board

BL0169 E-blocks II Mono LCD board

BL0138 E-blocks II Keypad board

BL0127 E-blocks II Actuators board

BL0129 E-blocks II Grove sensors board

BL0154 E-blocks II SD card board

BL0173 E-blocks II Terminal board

BL0162 E-blocks II Servo motor board

BL0152R E-blocks II Zigbee router board

BL0152C E-blocks II Zigbee coordinator board

BL0183 E-blocks II Relay board

BL0157 E-blocks II Colour graphical display board

BL0170 E-blocks II Bluetooth board

BL0136 E-blocks II Wifi board

BL0106 E-blocks II Mikroe adaptor board

BL0187 E-blocks II Ethernet board

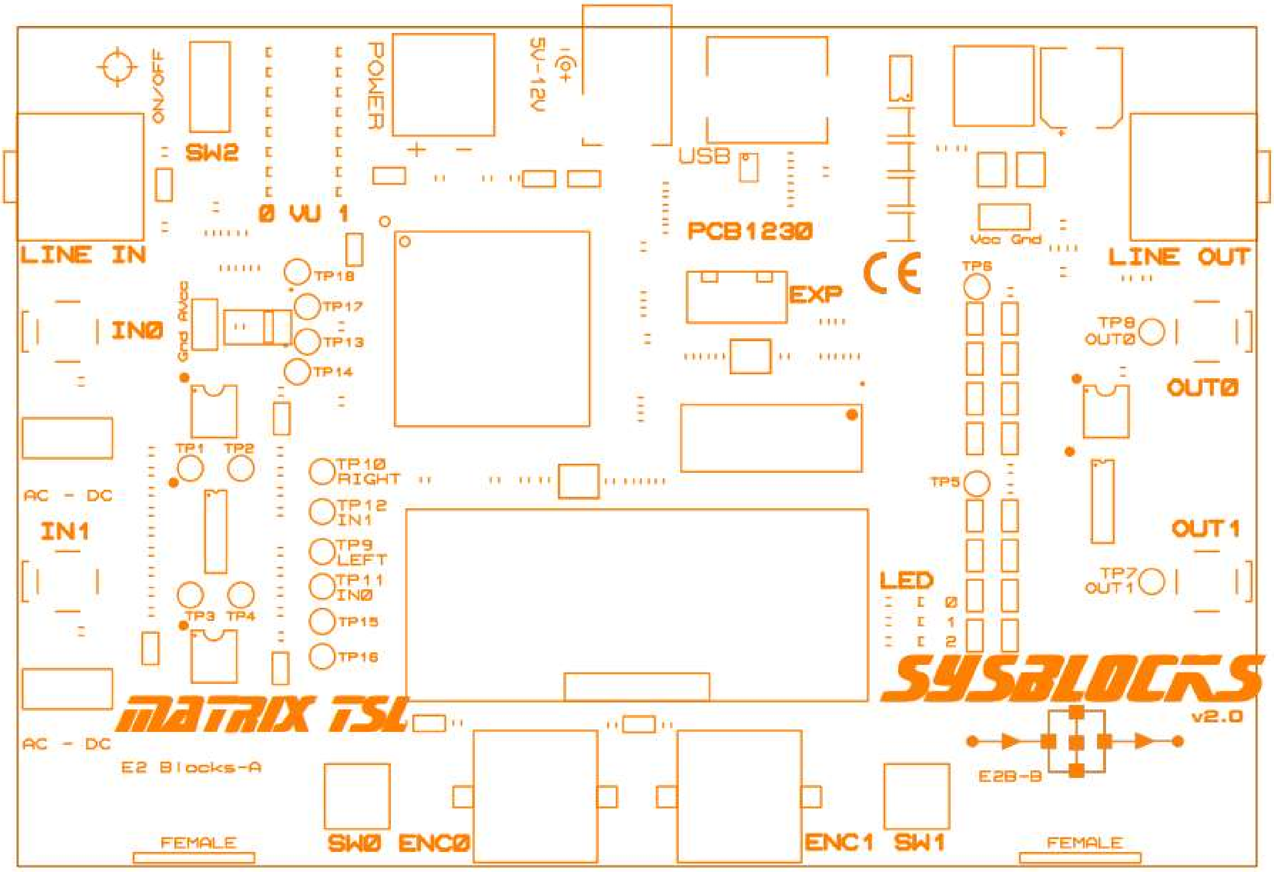
BL0169 E-blocks II Mono graphical LCD board

BL0140 E-blocks II CAN bus board

BL0197 E-blocks II RFID board

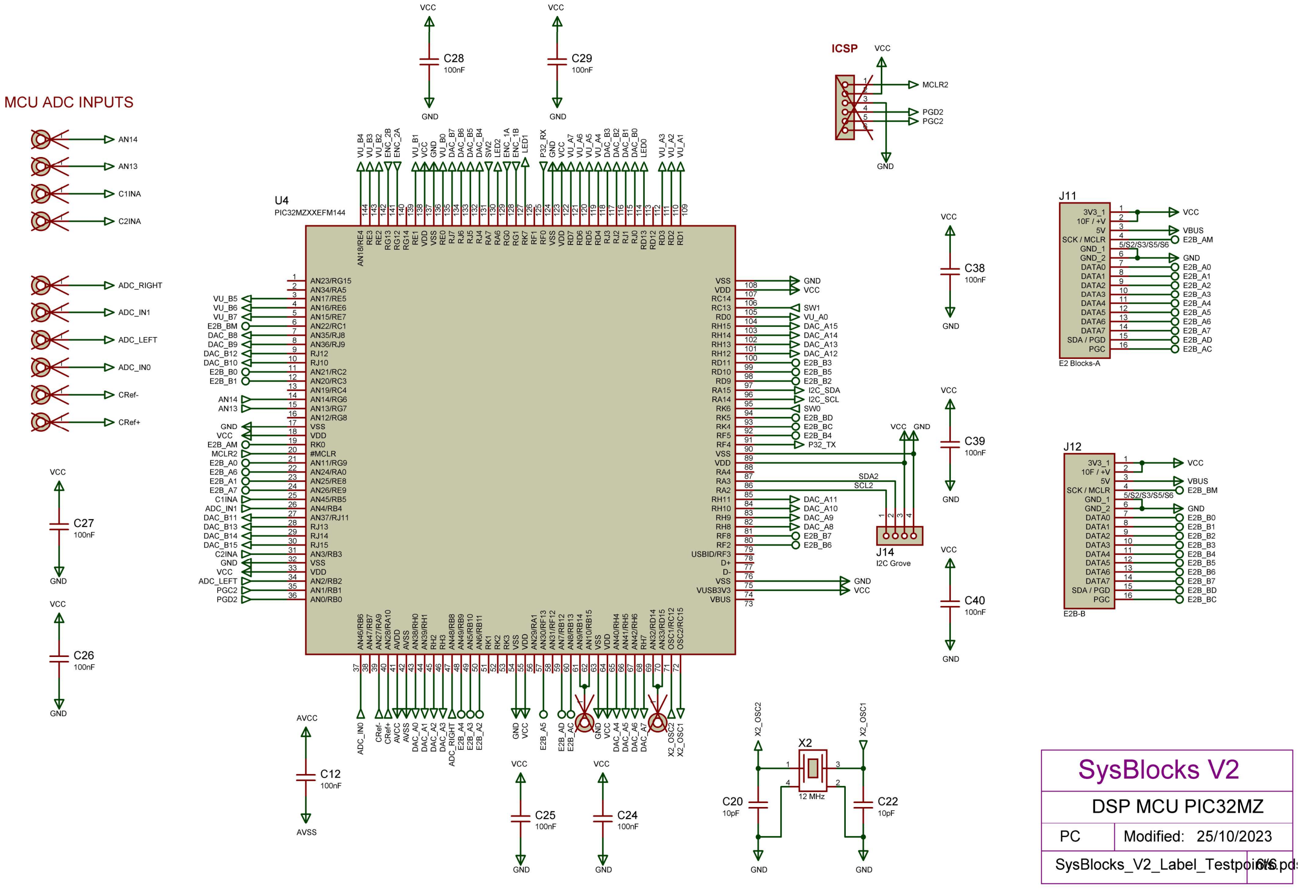
BL0131 E-blocks II GSM board

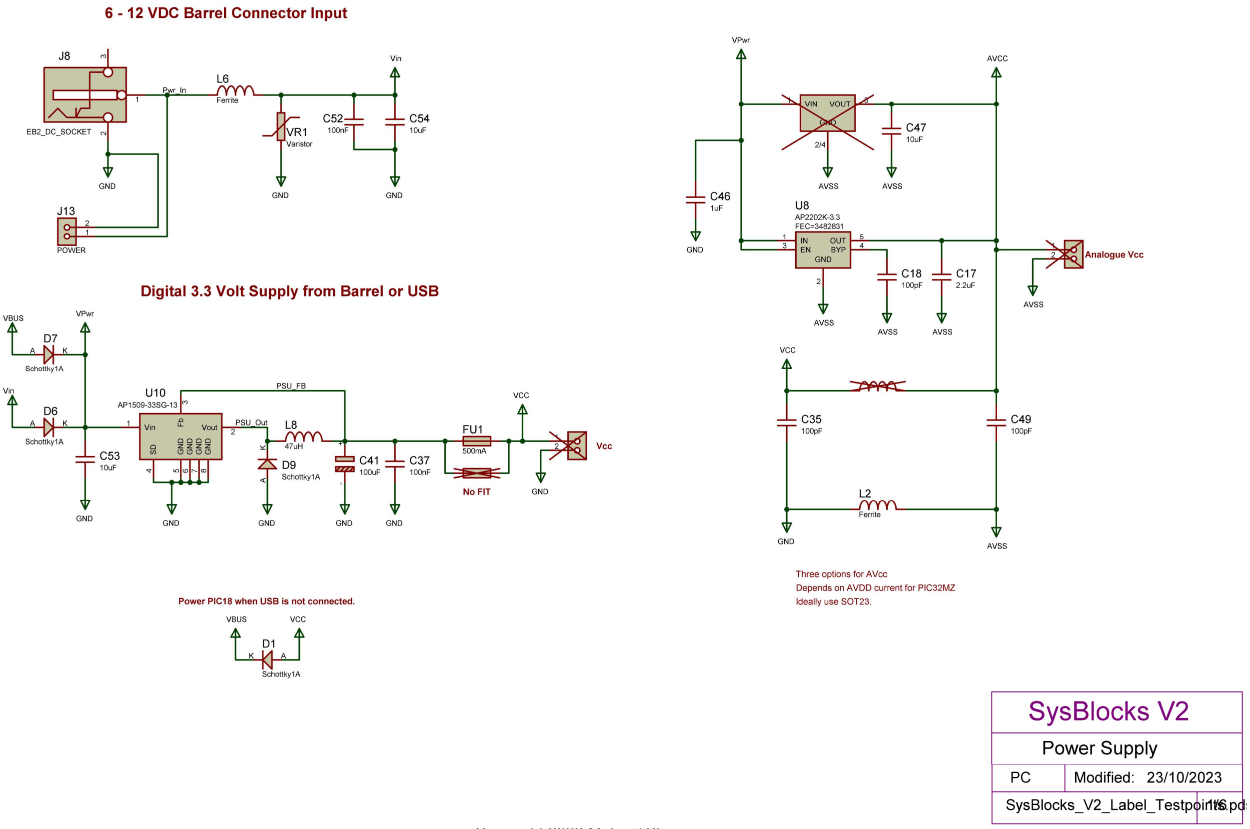
PCB layout



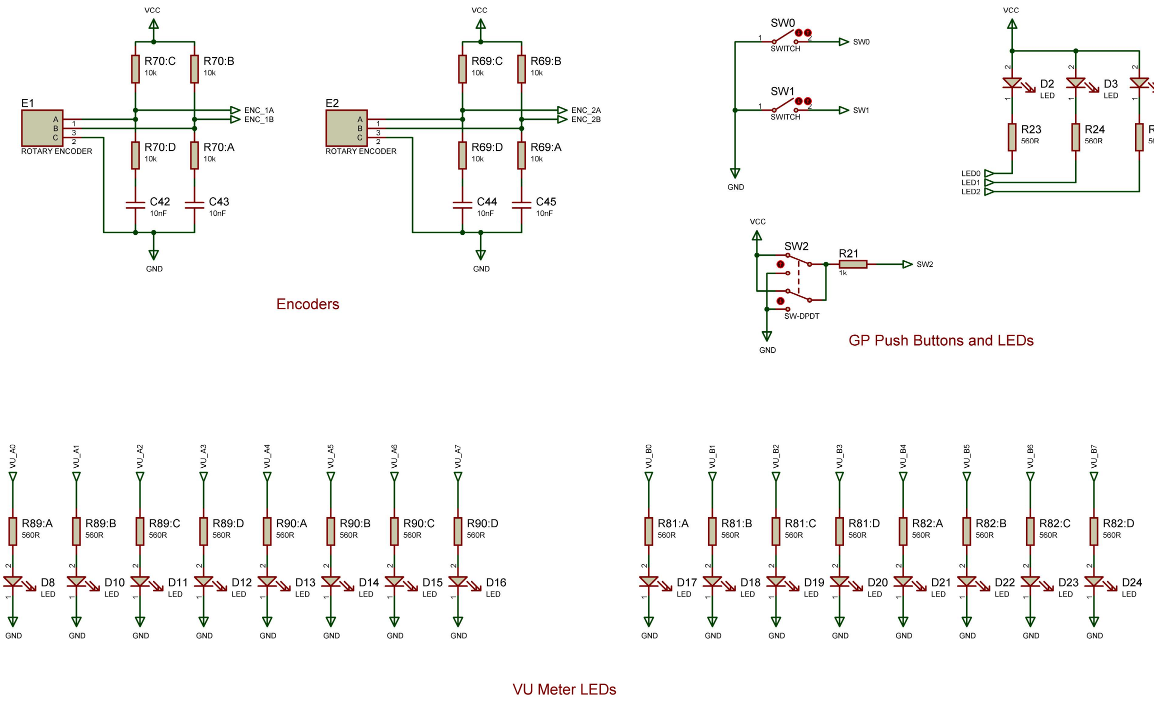
ON/OFF
SW2
POWER
5V-12V
USB
LINE IN
0 VU 1
PCB1230
Vcc Gnd
LINE OUT
TP6
TP18
TP17
TP13
TP14
IN0
Gnd Avcc
EXP
TP8 OUT0
OUT0
TP1 TP2
TP10 RIGHT
TP5
AC - DC
IN1
TP12 IN1
TP9 LEFT
TP11 IN0
TP15
TP16
TP3 TP4
LED
0
1
2
OUT1
TP7 OUT1
SYSBLOCKS
v2.0
MATRIX TSL
AC - DC
E2 Blocks-A
E2B-B
FEMALE
SW0 ENC0
ENC1 SW1
FEMALE

Schematics


MCU ADC INPUTS
ICSP
U4
PIC32MZXXEFM144
J11
J12
J14
I2C Grove
X2
12 MHz
SysBlocks V2
DSP MCU PIC32MZ
PC
Modified: 25/10/2023


6 - 12 VDC Barrel Connector Input
Digital 3.3 Volt Supply from Barrel or USB
Power PIC18 when USB is not connected.
Three options for AVcc
Depends on AVDD current for PIC32MZ
Ideally use SOT23.
Analogue Vcc
SysBlocks V2
Power Supply
PC
Modified: 23/10/2023

Schematics

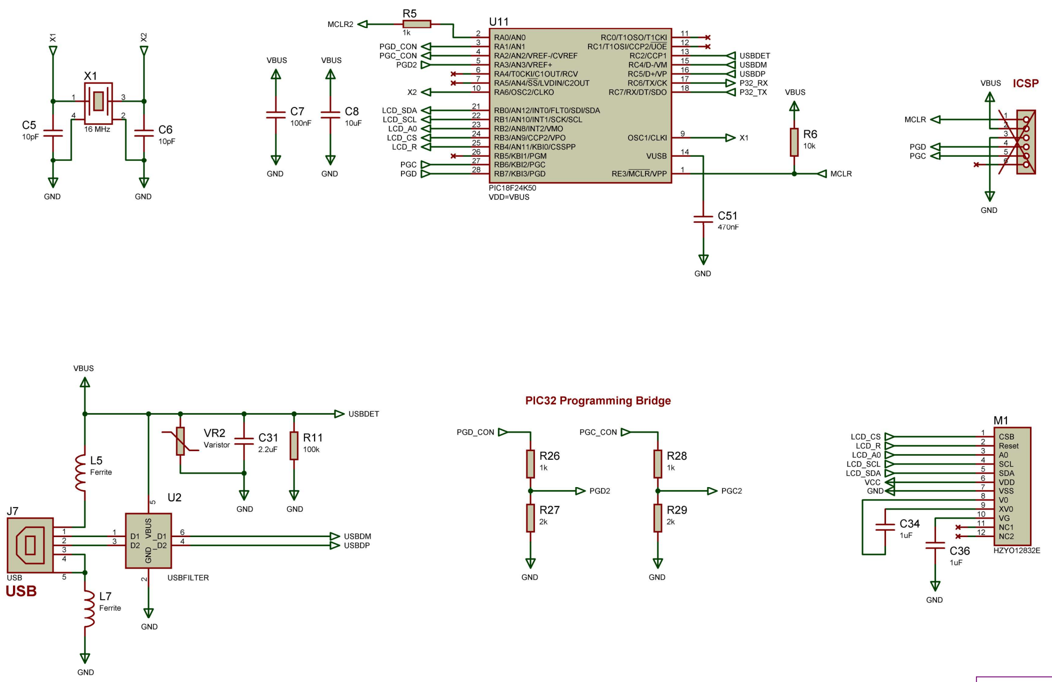


SysBlocks V2

User Interface

PC Modified: 23/10/2023

SysBlocks_V2_Label_Testpoints.pd 5/6



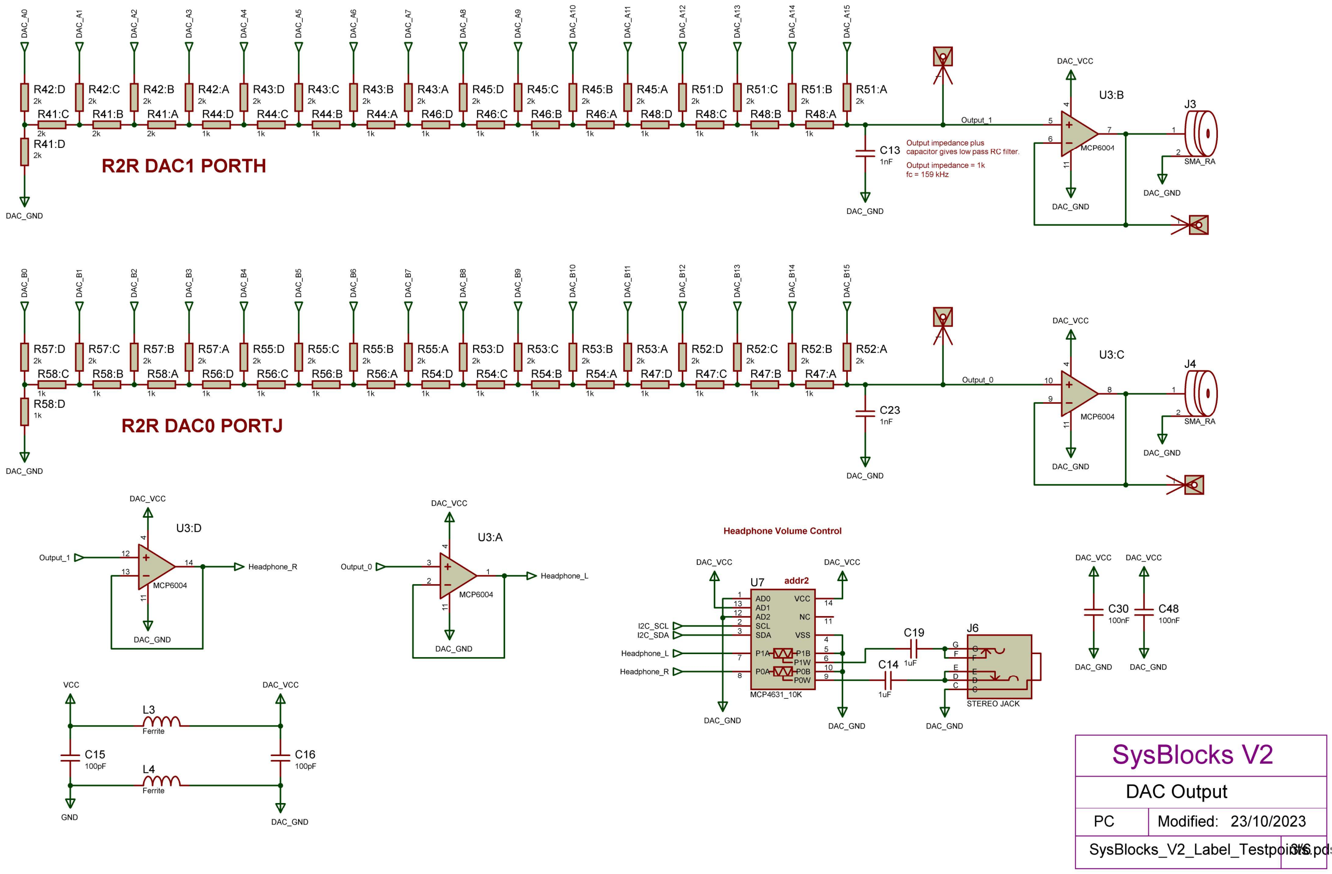
SysBlocks V2

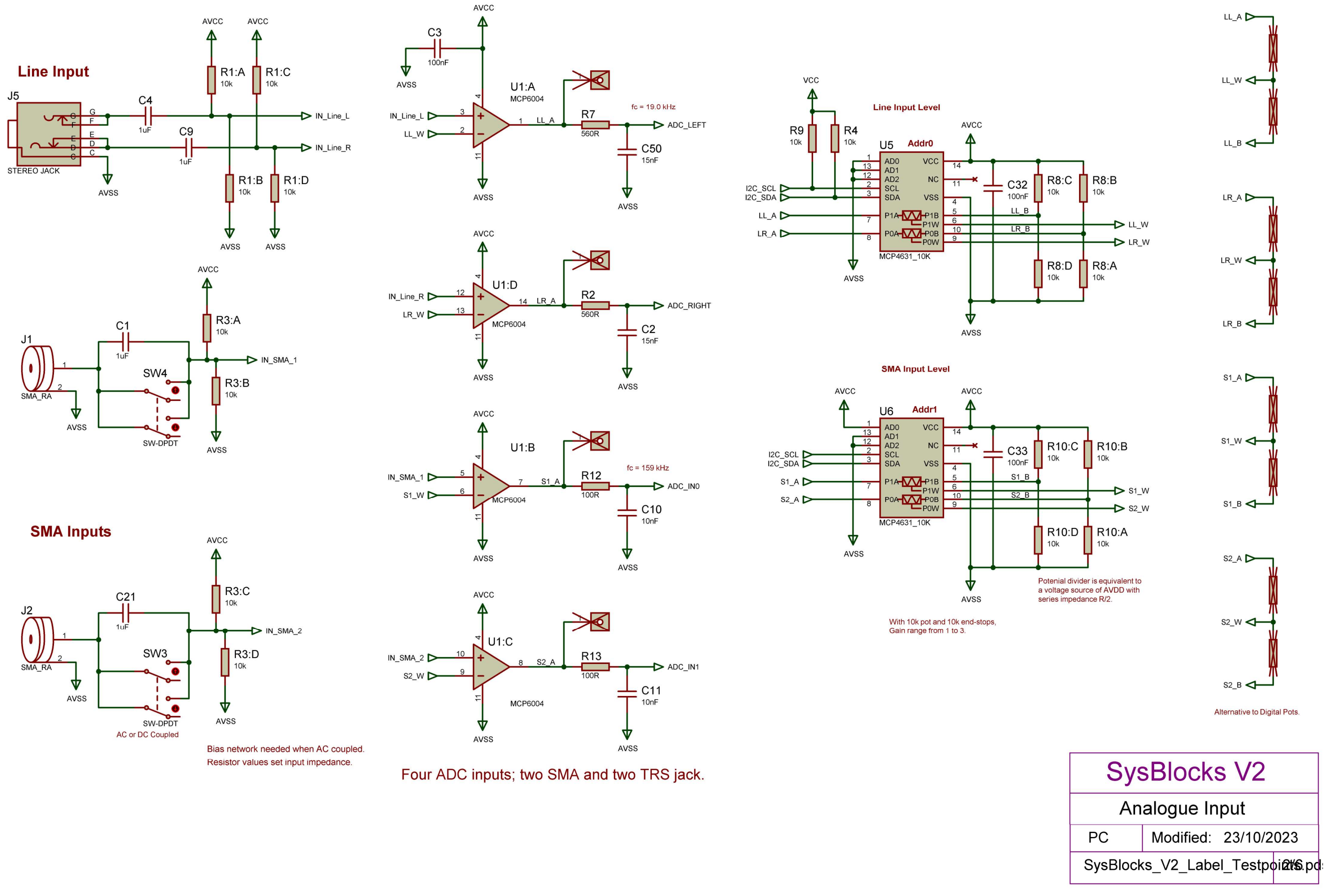
USB and LCD MCU

PC Modified: 23/10/2023

SysBlocks_V2_Label_Testpoints.pd 4/6

Schematics


R2R DAC1 PORTH
R2R DAC0 PORTJ
Output impedance plus capacitor gives low pass RC filter.
Output impedance = 1k
fc = 159 kHz
Headphone Volume Control
SysBlocks V2
DAC Output
PC
Modified: 23/10/2023
SysBlocks_V2_Label_Testpoints.pd


Line Input
SMA Inputs
Bias network needed when AC coupled.
Resistor values set input impedance.
AC or DC Coupled
Four ADC inputs; two SMA and two TRS jack.
Line Input Level
SMA Input Level
Potenial divider is equivalent to a voltage source of AVDD with series impedance R/2.
With 10k pot and 10k end-stops, Gain range from 1 to 3.
Alternative to Digital Pots.
SysBlocks V2
Analogue Input
PC
Modified: 23/10/2023
SysBlocks_V2_Label_Testpoints.pd

SYSBLOCKS

FLOWCODE
Electronic system design software

E BLOCKS 2

Matrix TSL

33 Gibbet Street

Halifax

West Yorkshire

UK

+44 1422 252380

www.matrixtsl.com