

Digital Filtering

TD 8: SigmaDSP - advanced functions

Antonin NOVAK

Laboratoire d'Acoustique de l'Université du Maine, Le Mans, France





1 General Purpose Input/Output

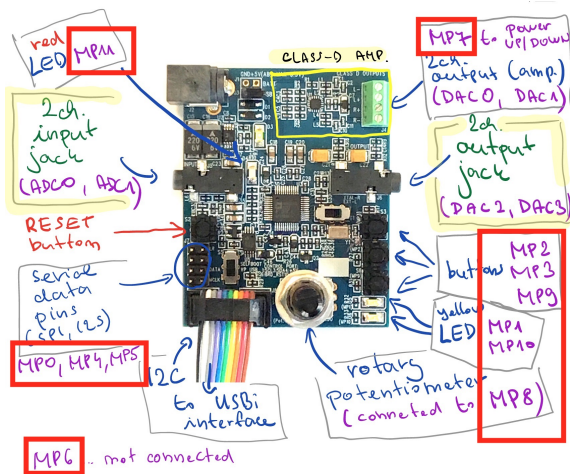
Hierarchy board

Controlling DSP with a μ -controller
using GPIO

Controlling DSP with a μ -controller
accessing memory using I2C

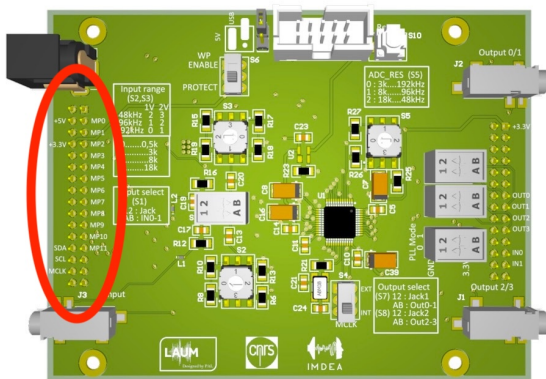
General Purpose Input/Output (GPIO)

EVAL-ADAU1701MINIZ



General Purpose Input/Output (GPIO)

LAUM-BOARD



General Purpose Input/Output (GPIO)

EVAL-ADAU1701MINIZ

Table 4. GPIO Setup

ADAU1701 Pin	Device	Settings	SigmaStudio Setting
MP1	Yellow LED D5	Active high	Output GPIO
MP2	Push-button S4	Push to ground	Input GPIO debounce, invert
MP3	Push-button S3	Push to ground	Input GPIO debounce, invert
MP7	SSM2306 SD	Active low	Output GPIO
MP8	10 kΩ potentiometer	Clockwise turn lowers resistance	ADC3
MP9	Push-button S6	Push to ground	Input GPIO debounce, invert
MP10	Yellow LED D6	Active high	Output GPIO
MP11	Red LED D4	Active high	Output GPIO

GPIO

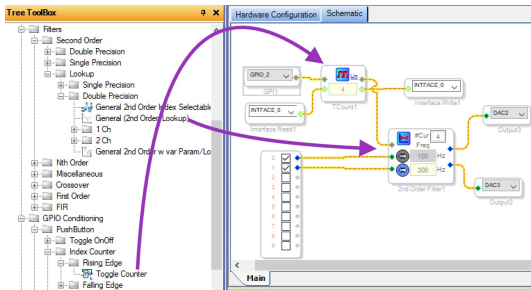
☐ Forced By SPI Debounce 20ms

Pin	Value	Direction	Inv
MP0	Low	NA	☐
MP1	Low	Output GPIO	☐
MP2	Low	Input GPIO Debounce	☒
MP3	Low	Input GPIO Debounce	☒
MP4	Low	NA	☐
MP5	Low	NA	☐
MP6	Low	NA	☐
MP7	Low	NA	☐
MP8	Low	ADC3	☐
MP9	Low	Input GPIO Debounce	☒
MP10	Low	Output GPIO	☐
MP11	Low	Output GPIO	☐

Work 1: Toggle filters with buttons

Use a button on the circuit board to toggle between predefined filters

1. In Sigma Studio create a new project and follow the schema below.



2. Click on "Filter" and prepare four filters
3. Connect a music and headphones and test it.
4. Save this project, you will need it later.

Use the Volume Control connected to buttons on the board

-
- The screenshot displays the Proteus ISIS software interface for hardware configuration. On the left, the 'Tree Toolbox' lists various components categorized under 'IO', including 'GPIO'. The 'Hardware Configuration' window on the right shows a circuit diagram with components like 'GPIO_2', 'GPIO_3', 'INTERFACE_0', 'DAC2', and 'DAC3'. The 'Main' window at the bottom shows a 'Capture' button. Purple arrows highlight the connections between the 'GPIO' component in the toolbox and the 'GPIO_2' and 'GPIO_3' components in the circuit diagram.

2. Click on "Table" on the Volume Control block and set the volume values from 0 to 1.
3. In Register Control (Hardware Configuration) set MP2 and MP3 as shown on slide 5.
4. Connect a music and headphones and test the volume control.



General Purpose Input/Output

2

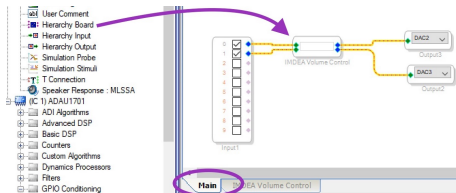
Hierarchy board

Controlling DSP with a μ -controller
using GPIO

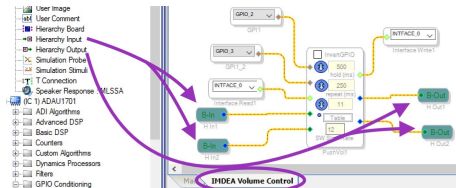
Controlling DSP with a μ -controller
accessing memory using I2C

Work 3: Hierarchy board

1. In Sigma Studio, copy/cut (ctrl+x) all previous work blocks except audio in / out.
2. Put a Hierarchy board instead.



3. The new Hierarchy board appears as a new Tab on bottom. Paste (ctrl+c) all the copied blocks and add Hierarchy inputs and outputs.





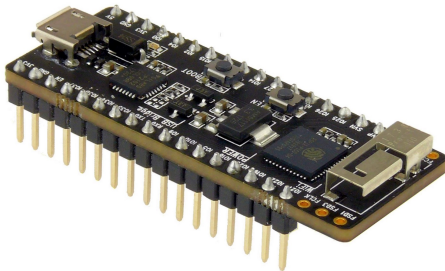
General Purpose Input/Output

Hierarchy board

3 Controlling DSP with a μ -controller
using GPIO

Controlling DSP with a μ -controller
accessing memory using I2C

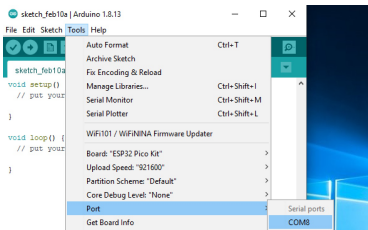
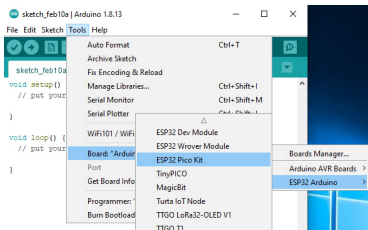
A **microcontroller** is a computer system on a chip that can be **programmed** and that is used to **interact** with things connected to the chip.



example of a board (ESP32 Pico) with a microcontroller ESP32

To program it, you will need **Arduino IDE** with support for ESP32
(all installed in the **Dirac room**).

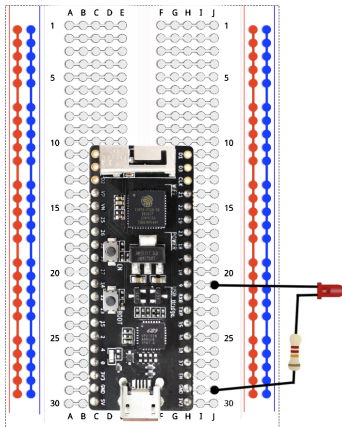
- Connect your ESP32 board and take a look on Windows task tray to note the number of port (e.g. COM3) while connecting.
- In Arduino IDE, Select the Board "ESP32 Pico Kit" and the port you have noted in previous step (see picture below).



Work 4: Test the ESP32 with a diode

Test the correct working of the ESP32.

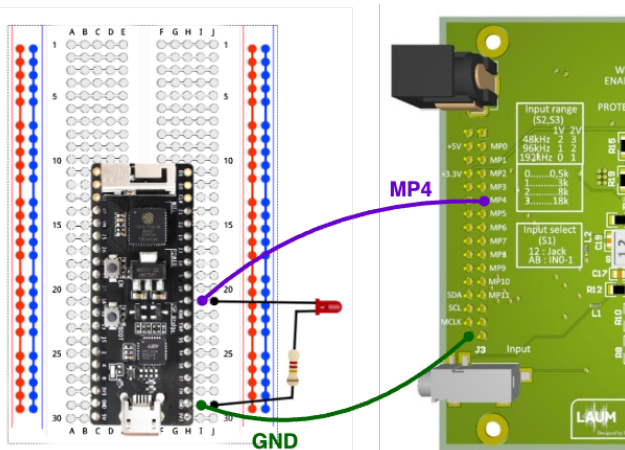
1. Download [TD08_esp32.zip](#) from umtice.
2. Connect a diode with a resistor as depicted below (GND, pin 9).
3. Open the project [test_01_LED.ino](#) and run it.



Work 5: ESP32 → DSP (GPIO)

Replace the button from Work 1 by ESP32 output pin (9).

1. Open the Sigma Studio project (work 1) and change the GPIO input to MP4.
2. Connect the DSP with ESP32 as depicted below (GND to GND , pin 9 to LR (MP4)).
3. Compile the Sigma Studio project and listen to it.





General Purpose Input/Output

Hierarchy board

Controlling DSP with a μ -controller
using GPIO

4

Controlling DSP with a μ -controller
accessing memory using I2C

All parameters of Sigma Studio project (filter coefficients, gains, ...) are saved in the memory of SigmaDSP.

When modifying these parameters in Sigma Studio, we can find at which address of the memory these parameters are saved.

They are shown in Capture View (Ctrl+5) (project need to be saved and compiled).

The screenshot displays the Sigma Studio interface. The top panel shows the 'Hardware Configuration' tab with a 'General Filter Settings' section. Under 'IIR Coefficient', there are five input fields: b0 (1.000000000000), b1 (0.200000000000), b2 (0.100000000000), a1 (-0.600000000000), and a2 (0.500000000000). The a1 and a2 fields are highlighted with a red box. Below these fields is a 'load' button. The main schematic area shows a 'Gen Filter' block connected to a 'Gain' block. The 'Gain' block has a '0.61' value and is connected to two DACs, DAC2 and DAC3. The bottom panel shows the 'Capture' view with a table of parameters and their memory addresses.

gain values (stereo)
memory address:
0x0005
0x0006

IIR filter,
coefficients a1, a2
with flipped sign

Mode	Time	Parameter Name	Address	Value	Data	Bytes
Safeload Write	16:16:10 - 116ms	Gen Filter1	EQ1940Dual10B1	0x0000	0x00, 0x80, 0x00, 0x00	4
Safeload Write	16:16:10 - 116ms	Gen Filter1	EQ1940Dual11B1	0x0001	0x00, 0x19, 0x99, 0x9A	4
Safeload Write	16:16:10 - 116ms	Gen Filter1	EQ1940Dual12B1	0x0002	0x00, 0x0C, 0xC, 0xCD	4
Safeload Write	16:16:10 - 116ms	Gen Filter1	EQ1940Dual13A1	0x0003	0x00, 0x4C, 0xC, 0xCD	4
Safeload Write	16:16:10 - 116ms	Gen Filter1	EQ1940Dual12A1	0x0004	0xFF, 0xCD, 0x00, 0x00	4
Block Write	16:16:12 - 650ms	Gain1	Gain1940A1gNS1	0x0005	0x00, 0x4E, 0x14, 0x7B	4
Block Write	16:16:12 - 650ms	Gain1	Gain1940A1gNS2	0x0006	0x00, 0x4E, 0x14, 0x7B	4

Work 6: ESP32 → DSP (I2C)

Communicating with SigmaDSP using I2C protocol

1. Create a Sigma Studio project with one 2-channel General 2nd order IIR filter (IIR Coefficients) followed by a stereo gain (see previous slide).
2. Connect the DSP and the USBi programmer with ESP32 as depicted below.
3. Open the project `test_02_ADAU_I2C.ino` and run it.

