ADIUVO
ENGINEERING AND TRAINING, LTD.

Intel-FPGA as a Companion Chip

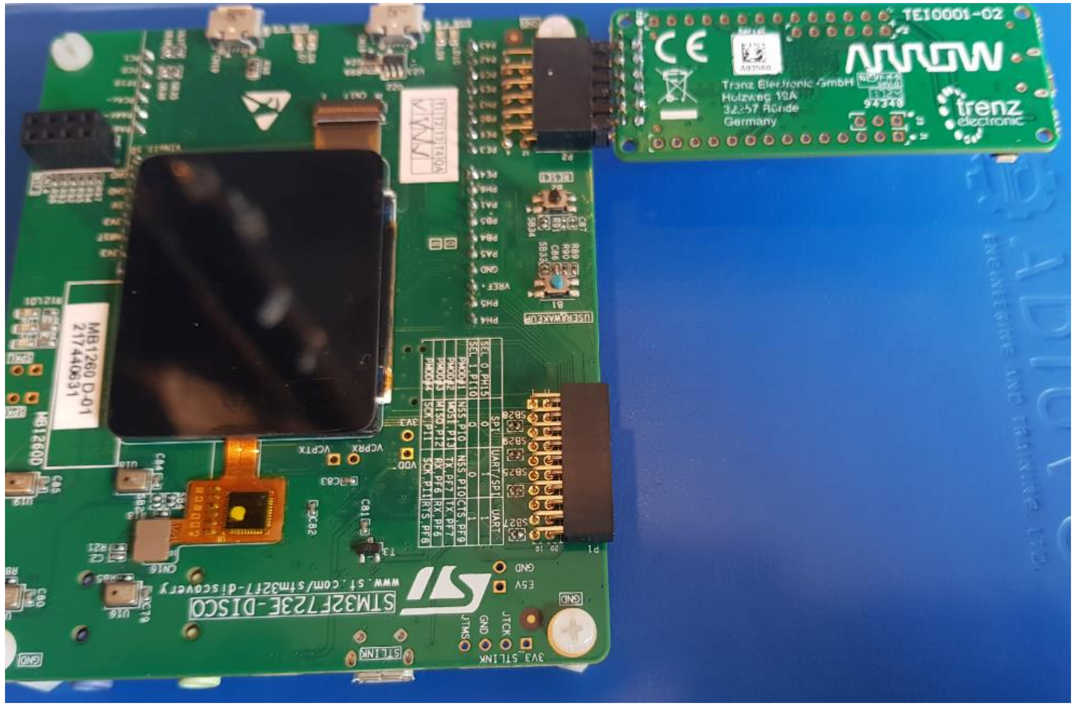
Adam Taylor

Objective

To enable a traditional microprocessor to be able to access non-standard interfaces.

Use a low-cost Intel-FPGA board to translate from a processor supported interface e.g. SPI, I2C, QSPI, OSPI to one which is not for example analogue, GPIO, Motor Control, Cameras, Displays etc.

For this example, we are using Max1000 board and STM32F723E Disco board



Why is a companion need

Several reasons a companion chip may be required, driven by system and interfacing requirements.

- Interface Translation – Customized / Bespoke / legacy interface
- Interface Expansion – Providing additional standard interfaces
- Deterministic response – Motor control, Servo Positioning
- Offload signal processing – Filtering / noise removal of ADC signals
- Leverage parallelism in the Intel FPGA – Process multiple signals / sensors simultaneously.

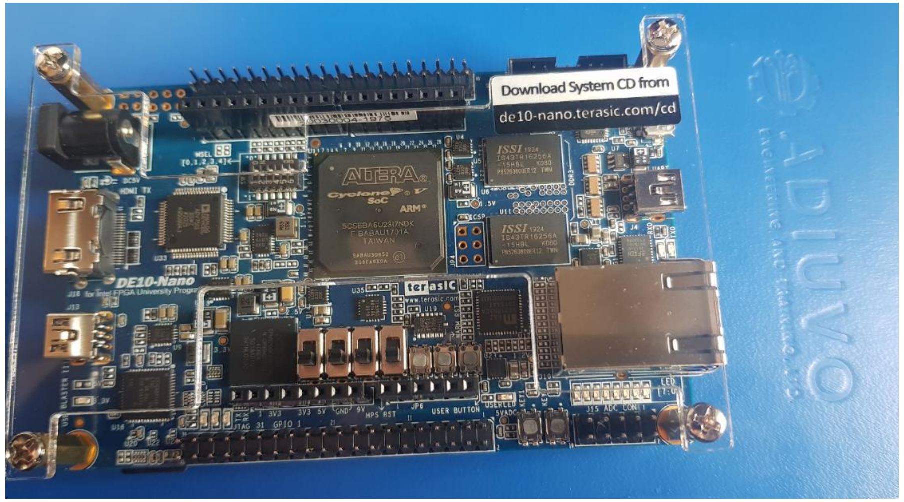
Why not use a SoC

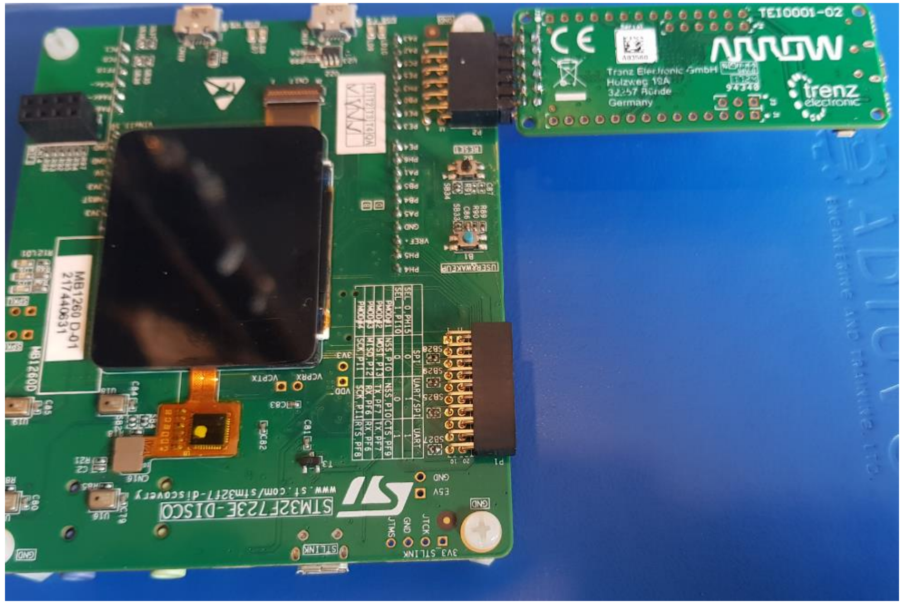
SoC combine Arm processors (A9, A53 etc.) with programmable logic.

Many developments may use a non-Arm processor and still need expansion for example Intel Atom

Logic resources might be very low depending on interface translation, making SoC overkill.

Ultimately at the end of the day the decision SoC or Intel FPGA and Companion board depends upon the system requirements.





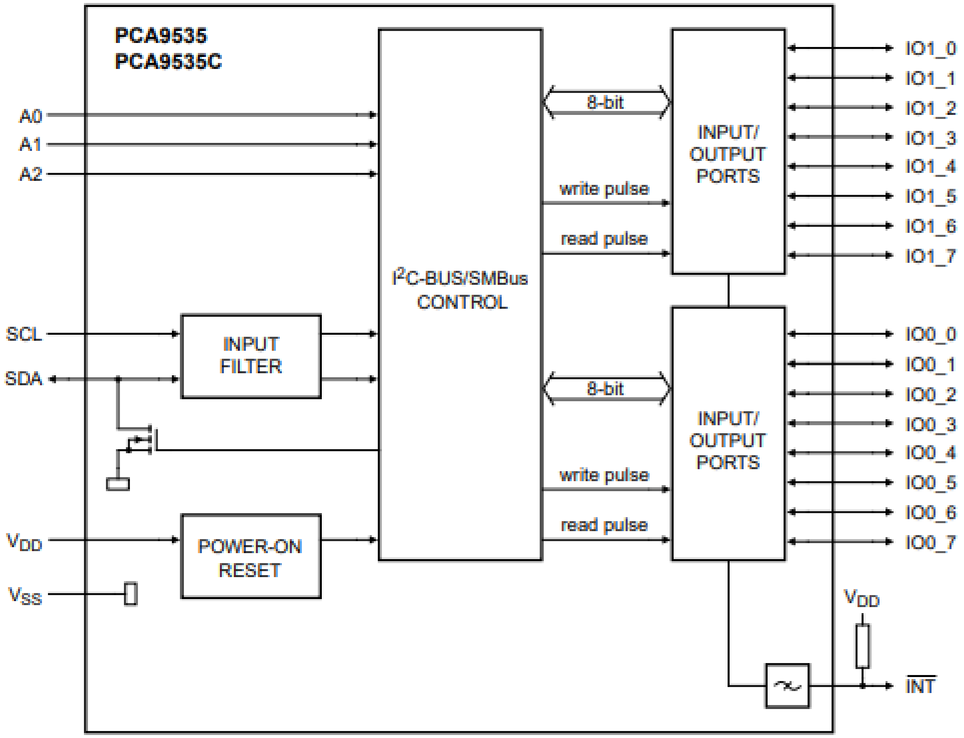
Why not use a simple IO Expander

Simple expanders for example I2C have no intelligence

Presented solution enables direct access to Avalon Master (via SPI/I2C)

Enables not only interfacing solutions but also

- Ability to monitor system health using inbuilt ADC
- Embedded signal processing – Filtering simple rolling average to FIR
    - Reduces processing load on the software – Software can work with the answer
    - Provides a more deterministic response
- Co-Processor could be used if desired for example



The Max1000 Board

Board is small footprint with flexible IO

Low-cost device

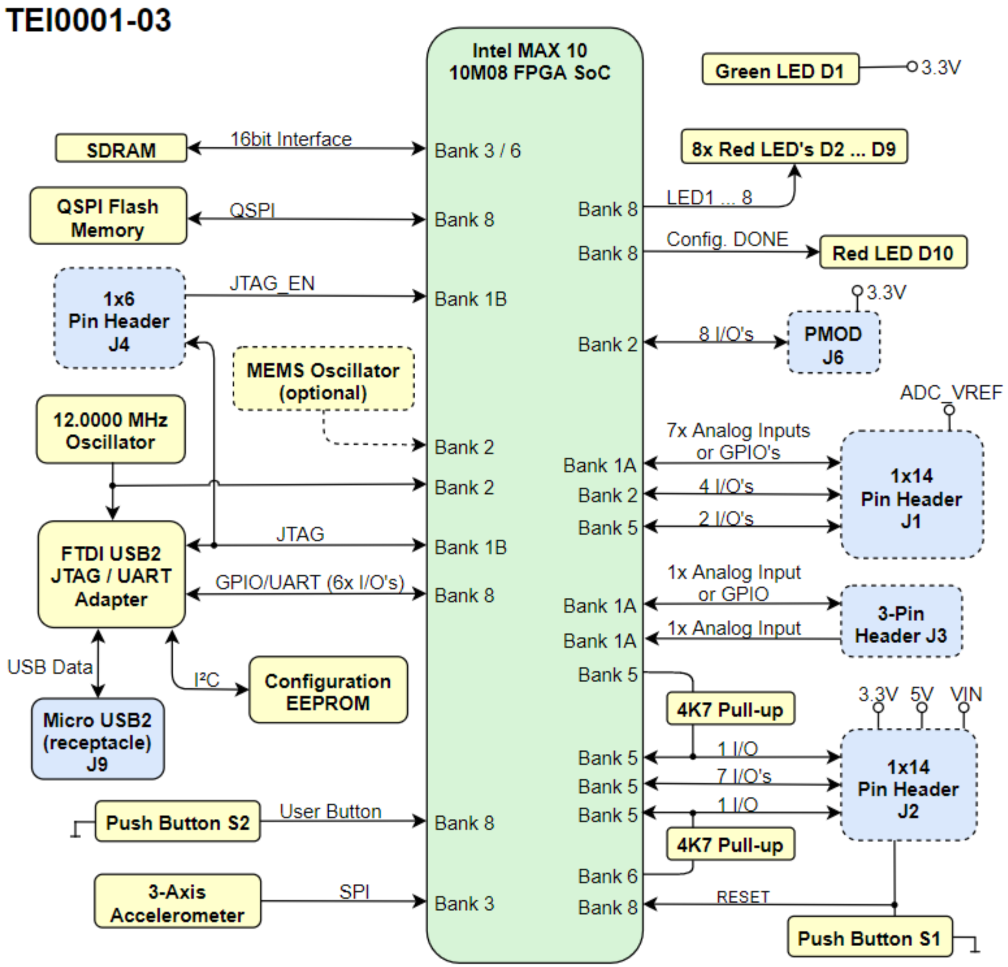
Low Overhead to include in design either board or Max10 Device

Flexible IO – provides a range of options

Breaks out most IO

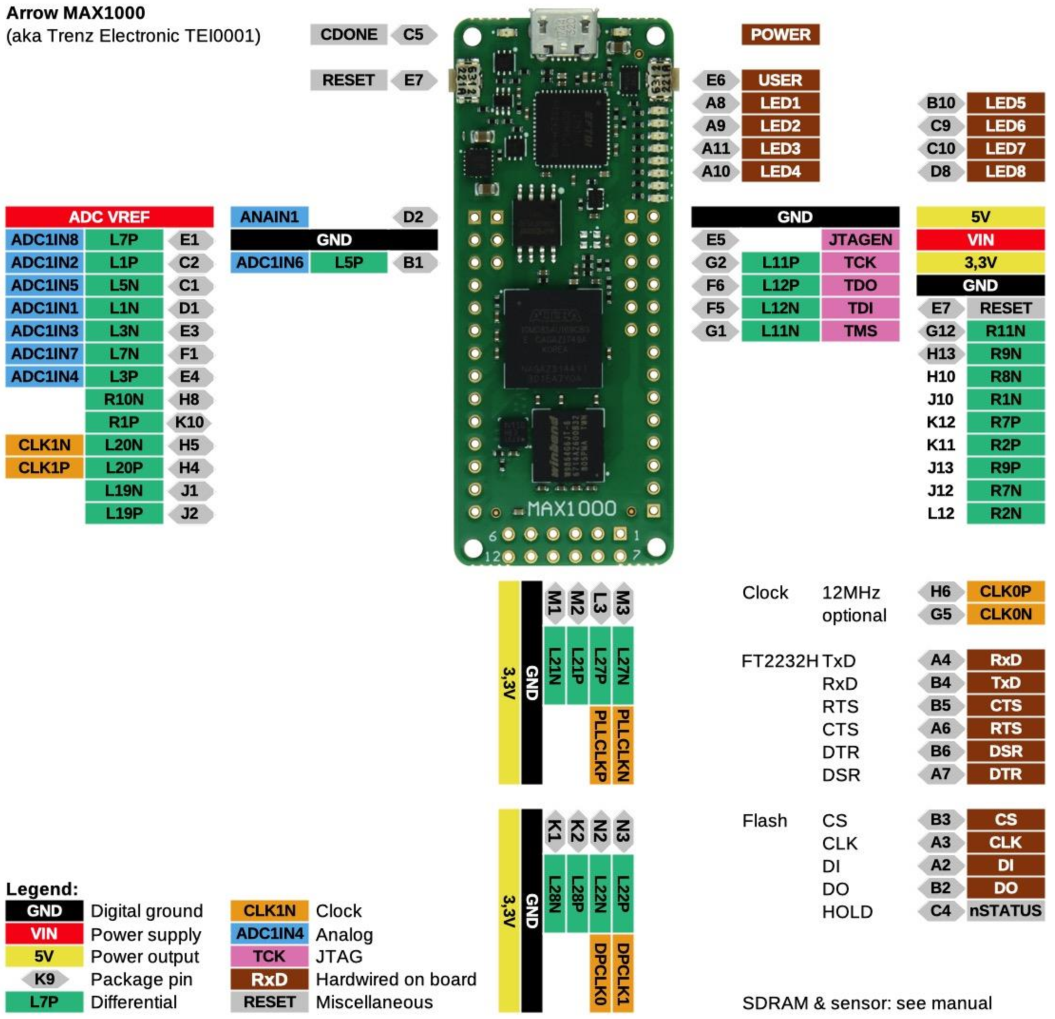
- Provides simple LED / SW
- ADC

Ideal for demonstration applications



Max1000 Pin Out

Pmod is master – Important as 3v3 is output not input – Be careful if connecting to another Pmod Interface.



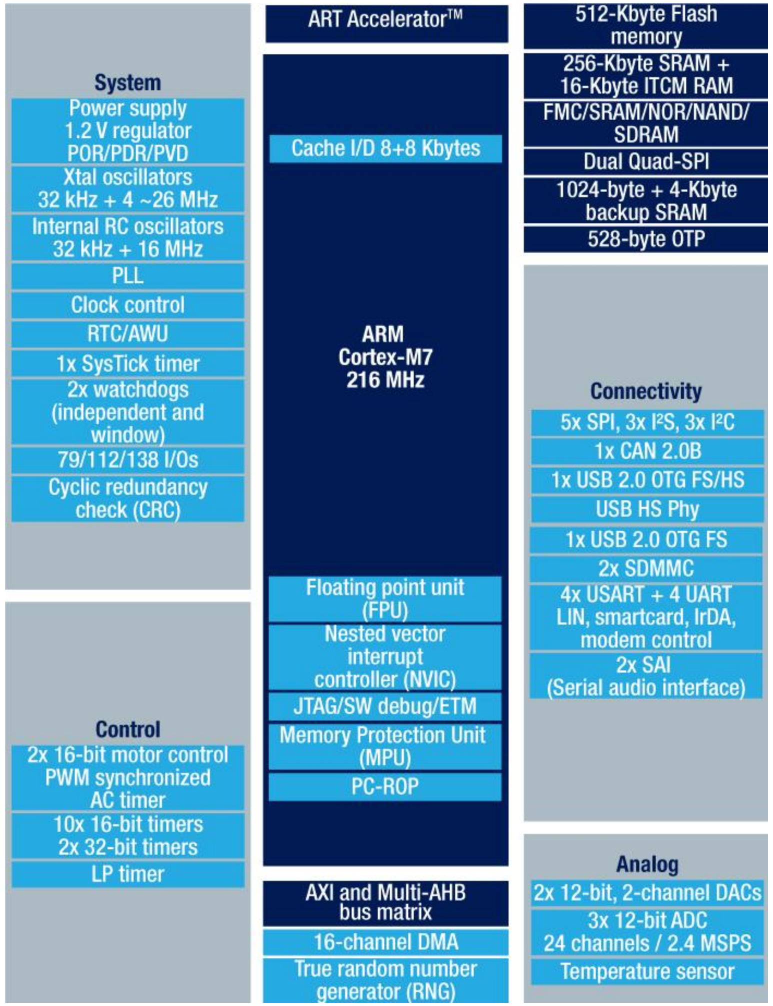
The STM32F723 Disco

Contains Arm Cortex M7 processor with FPU.

Very flexible protocol support with USB, CAN, SPI, I2C, I2S and UART

Connectivity Provides

- Arduino Shield
- Pmod
- Touch Screen
- Wifi
- On board St-Link and USB UART



I2C or SPI

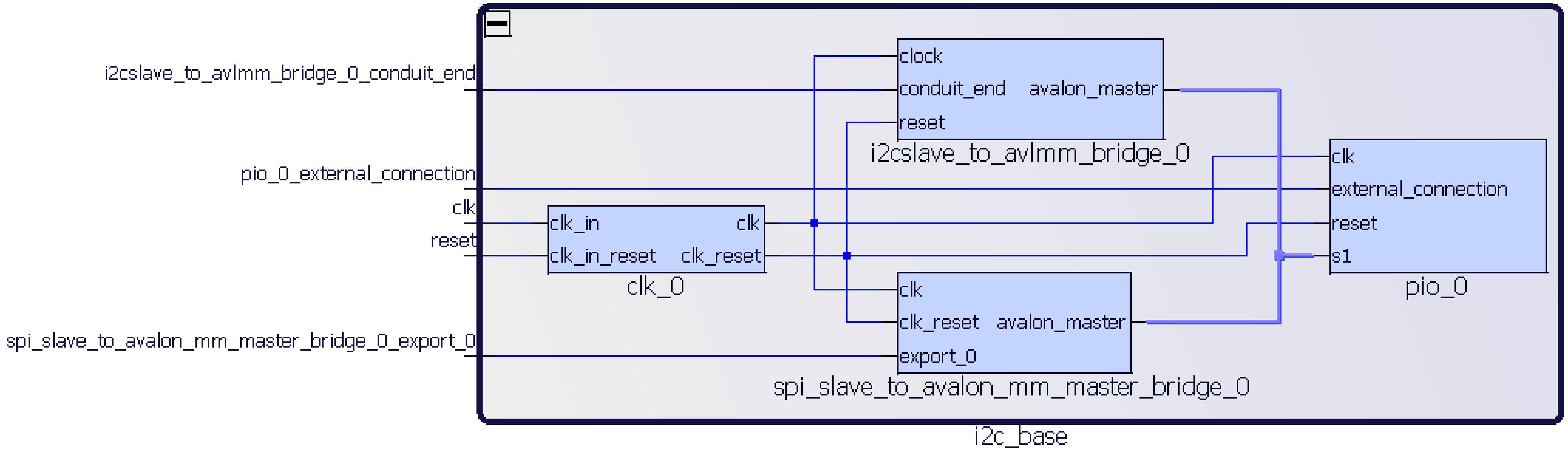
- Of course, depends on the interfaces available on the processor
- Both commonly provided in embedded systems
- SPI more complex and requires more lines
- I2C simpler – only two lines
- SPI higher performance – Full duplex
- I2C – Half Duplex
- SPI – Expansions for Quad and Octal to achieve much higher data rates
- I2C – Multi Master Support – Multiple Masters could access the Companion Chips

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- 55 -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Solution Architecture

Architecture of the design provides multiple Avalon Master access points.

I2C and SPI slave can access the Avalon bus (via Interconnect) to access the implemented slave peripherals



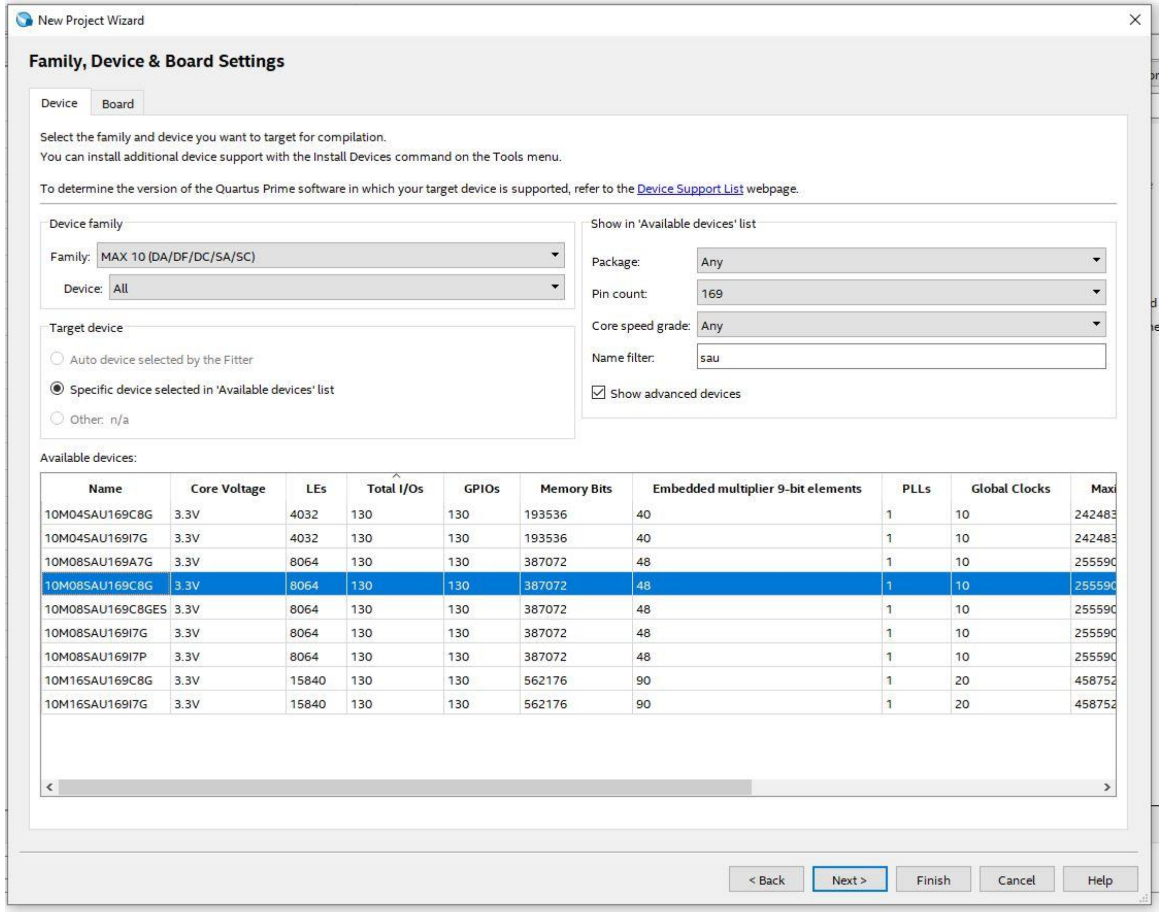
Note, SPI and I2C in same design is for reference only

Quartus Tool Chain

Design Created in Quartus, targeting Intel Max 10 device.

Development utilized Platform Designer – System Integration tool.

Debugging on target can be achieved using Signal Tap – In chip Logic Analyzer.



Platform Designer

Platform designer allows us to leverage existing IP

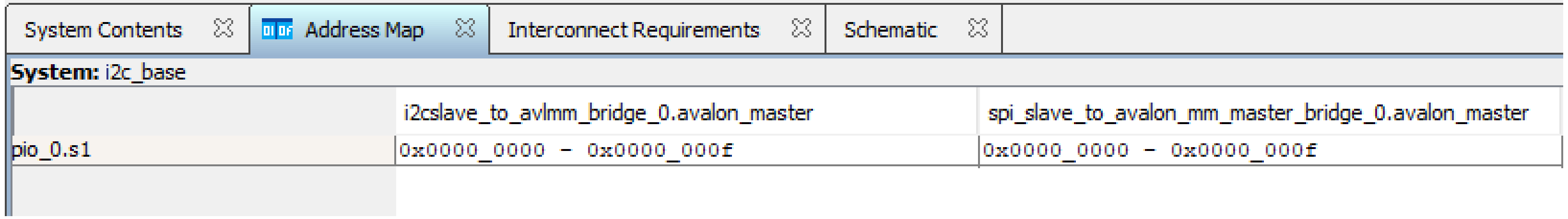
Custom IP can be created for easy integration to the Avalon MM bus

Use	Connections	Name	Description	Export	Clock	Base
☑		⊟ **clk_0**	Clock Source			
		clk_in	Clock Input	**clk**	***exported***	
		clk_in_reset	Reset Input	**reset**		
		clk	Clock Output	*Double-click to export*	clk_0	
		clk_reset	Reset Output	*Double-click to export*		
☑		⊟ **i2cslave_to_avlmm...**	I2C Slave To Avalon-MM Master Bridg...			
		clock	Clock Input	*Double-click to export*	**clk_0**	
		avalon_master	Avalon Memory Mapped Master	*Double-click to export*	[clock]	
		reset	Reset Input	*Double-click to export*	[clock]	
		conduit_end	Conduit	**i2cslave_to_avlmm_bridge_0_conduit_end**		
☑		⊟ **pio_0**	PIO (Parallel I/O) Intel FPGA IP			
		clk	Clock Input	*Double-click to export*	**clk_0**	
		reset	Reset Input	*Double-click to export*	[clk]	
		s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk]	0x0000_0000
		external_connection	Conduit	**pio_0_external_connection**		
☑		⊟ **spi_slave_to_avalon...**	SPI Slave to Avalon Master Bridge Int...			
		clk	Clock Input	*Double-click to export*	**clk_0**	
		clk_reset	Reset Input	*Double-click to export*	[clk]	
		export_0	Conduit	**spi_slave_to_avalon_mm_master_bridge_0_export_0**		
		avalon_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]	

Address Space

Address space can be optimized for the size of the peripheral being driven - no need to select 32 bit when only 8 bit addressing needed.

Selecting correct, address width will ensure the higher bandwidth as the transactions are more efficient.



Accessing Avalon MM over I2C & SPI

Explain read and write example – for both SPI and I2C

Software access at lowest level

Bare Metal – Hardware abstraction libraries

With Linux we can leverage frameworks like i2c-tools, Smbus and I2C and SPI Dev

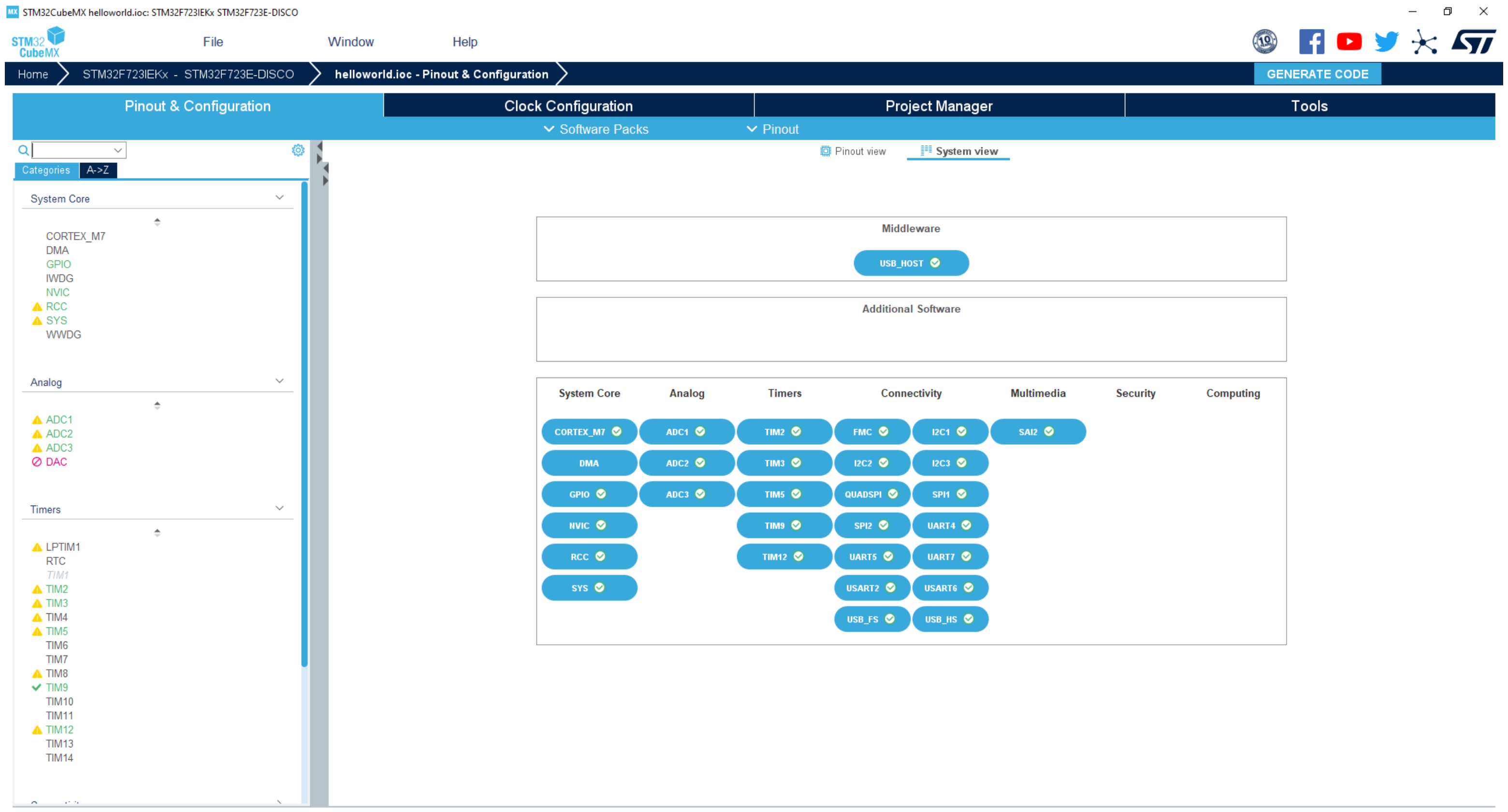
Make it very easy for Linux systems to be able to interact with the companion board

STM Software Solution

To create software solution for STM32F723 Discovery we need the following

STM Cube MX - Generates the low-level system configuration and start up code along with the project.

STM Cube IDE - IDE used for development of the application SW



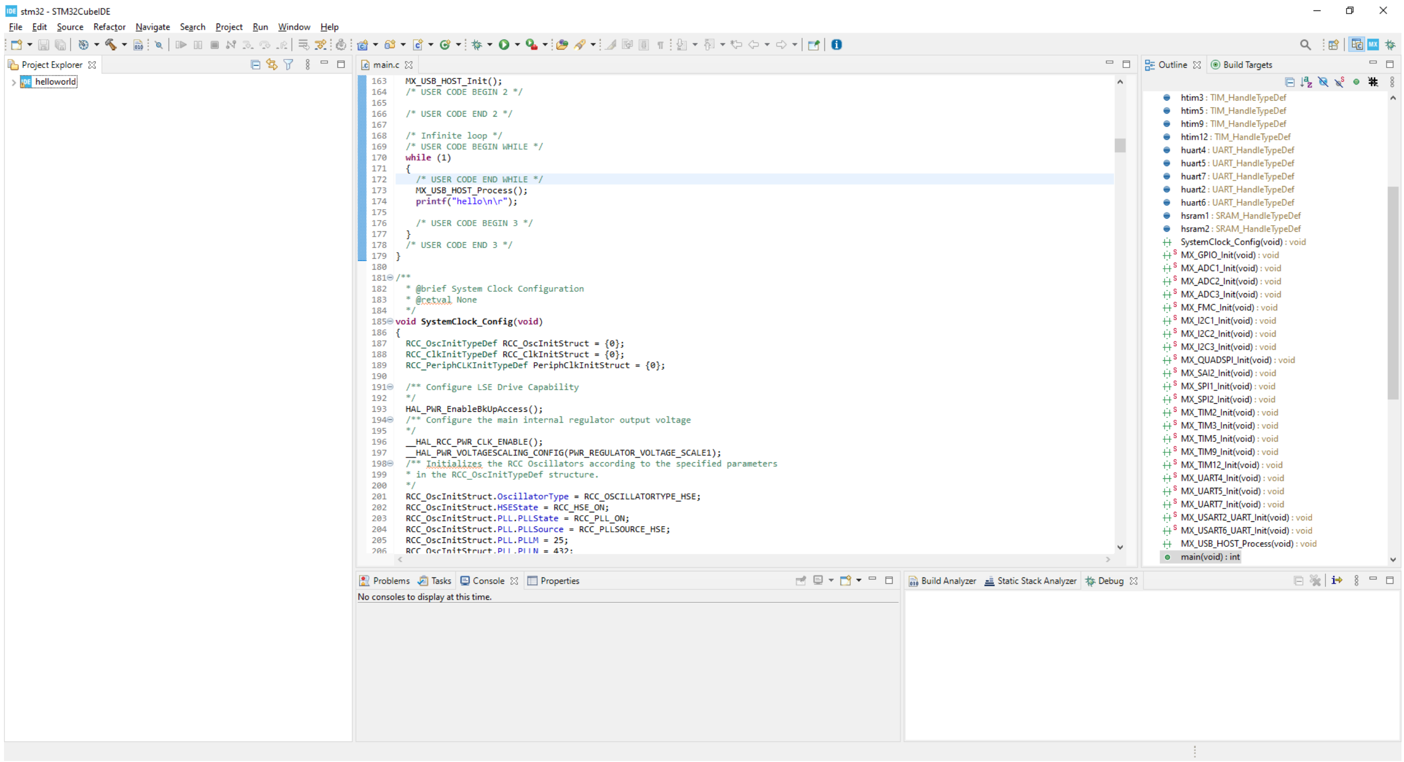
Software Development

Application development uses C.

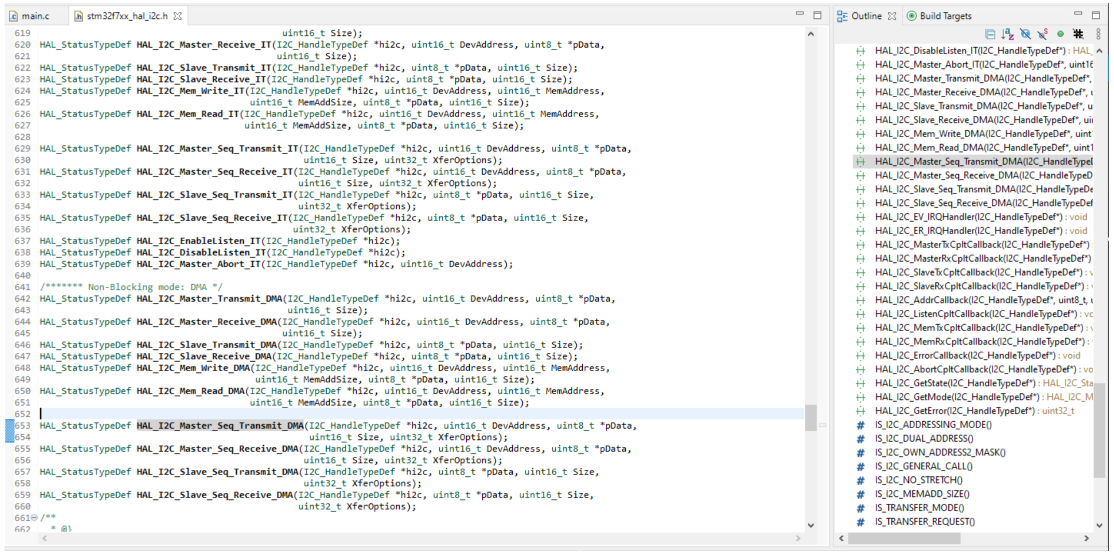
Access to Hardware via Hardware Abstraction Layer (HLA)

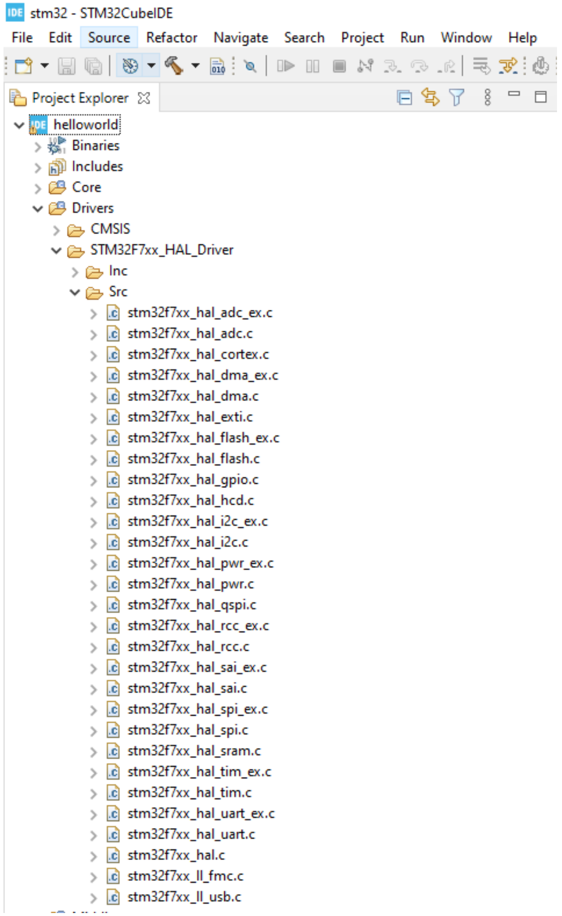
Debugging via ST-Link and GDB.

- » Single Step / Step Into
- » Breakpoints
- » Watch Values
- » Memory View etc



Hardware Abstraction Layer





Number of function calls which can be accessed from the main application to implement I2C or SPI transactions.

Demonstration

Wrap Up

Demonstrated a flexible method for creating a companion chip design expanding IO or supporting bespoke or new interfaces.

Demonstration outlined how to use I2C to break out to LED GPIO using the MAX1000 board - Can be expanded to support a range of IO

Creation of project for the Intel FPGA can be achieved in 30-60 minutes, with similar for SW development of the simple application software.

What do you want to do with your application ?

Call to Action

All instructions and files to recreate the project can be found at https://github.com/ATaylorCEngFIET/Intel_Max1000

MAX1000:

https://www.arrow.com/en/products/max1000/arrow-development-tools

STM32 Discovery Kit:
https://www.arrow.com/en/products/stm32f723e-disco/stmicroelectronics?q=STM32F723E-Disco

Questions ?

Something comes to mind after, the session email adam@adiuvoengineering.com

ADIUVO
ENGINEERING AND TRAINING, LTD.

www.adiuvoengineering.com

adam@adiuvoengineering.com