ARM XMC - experiment No. 4

- **E4.1 Name:**

Infineon ARM XMC Boot kit – car board instrumentation

- **E4.2 Overview and purpose**

The experiment explores hardware and software resources of XMC Infineon platform Boot kit and programming environment used to design DAVE active elements for on-board automotive computers / installations.

At the end of the experiment, you will have detailed information about how to make the instrument cluster interfacing their functions and implementing Infineon platform to display a parameter (speed, engine speed, etc.).

Figure 4.1 A car board display

- **E4.3 Resources**

Hardware	XMC1100 development board for Arduino , display car step by step motor, oscilloscope.
Software	DAVE 4.1

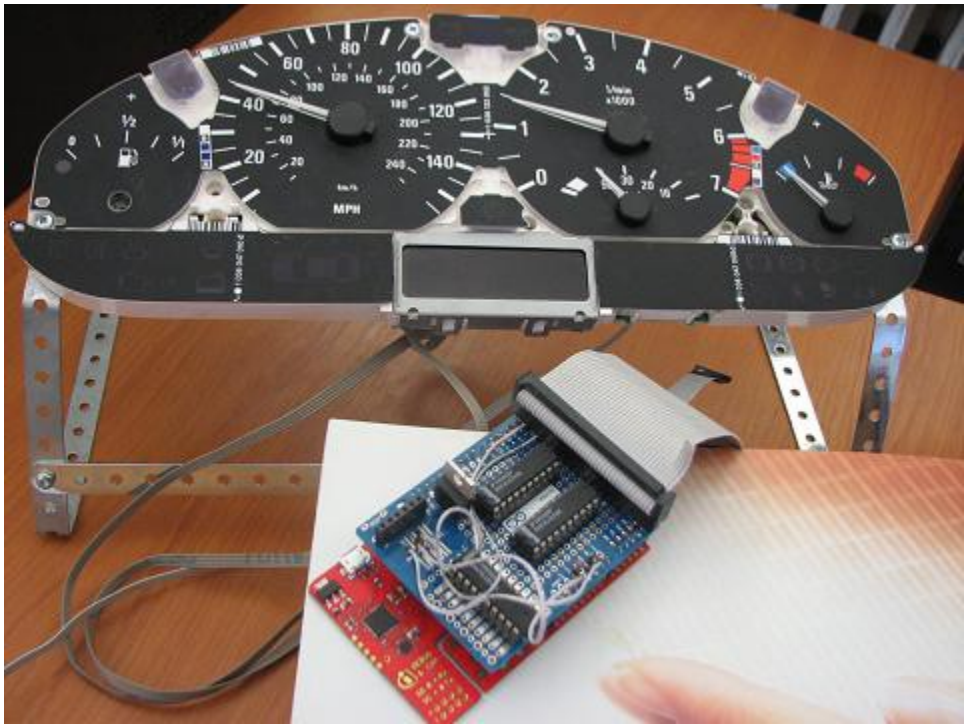

Figure 4.2 The eperiment example

- **E4.4 Software example:**

```
/*
Lab4 - Use XMC1100 to action a motor for instrumentation
Connection: coil A:  P1.4, P0.0
            coil B: P2.11, P2.10
 */

#include <DAVE.h> //Declarations from DAVE Code Generation (includes SFR declaration)

int j=6;
int k,l,t;

void delay(long unsigned int i) // delay
{
      while(i--)
      {
            j++;
            j--;
      }
}

/**
 * @brief main() - Application entry point
 *
```

```
 * <b>Details of function</b><br>
 * This routine is the application entry point. It is invoked by the device
startup code. It is responsible for invoking the APP initialization dispatcher
routine - DAVE_Init() and hosting the place-holder for user application code.
 */
int main(void)
{
  DAVE_STATUS_t status;

  status = DAVE_Init(); /* Initialization of DAVE APPs  */

  if(status == DAVE_STATUS_FAILURE)
  {
      /* Placeholder for error handler code. The while loop below can be
replaced with an user error handler. */
    XMC_DEBUG("DAVE APPs initialization failed\n");

    while(1U)
    {
      ;
    }
  }

   /* Placeholder for user application code. The while loop below can be
replaced with user application code. */
  DIGITAL_IO_SetOutputHigh (&a1);
  DIGITAL_IO_SetOutputHigh (&a2);
  DIGITAL_IO_SetOutputHigh (&b1);
  DIGITAL_IO_SetOutputHigh (&b2);
  t=4000;
  while(1U)
  {
       for(k=1;k<120;k++) //right rotation
       {
            DIGITAL_IO_SetOutputLow (&a2);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&a2); // current: i1

            DIGITAL_IO_SetOutputLow (&b2);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&b2); // current: i3

            DIGITAL_IO_SetOutputLow (&a1);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&a1); // current: i2

            DIGITAL_IO_SetOutputLow (&b1);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&b1); // current: i4
       }
       delay(50000);
       for(k=1;k<120;k++) // left rotation
       {
            DIGITAL_IO_SetOutputLow (&b1);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&b1); // current: i4
            DIGITAL_IO_SetOutputLow (&a1);
            delay(t);
```

```
            DIGITAL_IO_SetOutputHigh (&a1); // current: i2
            DIGITAL_IO_SetOutputLow (&b2);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&b2); // current: i3
            DIGITAL_IO_SetOutputLow (&a2);
            delay(t);
            DIGITAL_IO_SetOutputHigh (&a2); // current: i1
        }
        delay(50000);
    }
}
```

- **E4.5 Method of running experiment:**

1. Analyze user manual of XMC platform for Arduino (http://www.mouser.com/pdfdocs/InfineonBoard_Users_Manual_XMC1100_CPU_Card_R2.PDF) and identify signals compatible Arduino Uno connectors;
2. Explore technical documentation for step by step micro-motors (http://www.mikrocontroller.net/attachment/114256/switec_X15.xxx.02.SP.E.pdf) used for instrumentation panel.
3. Analyze XMC1100 scheme for Arduino platform;
4. Examine the interconnection scheme between micro-motors and XMC platform;
5. Configuring the output ports used to shareholder the micro-motors;
6. Load the prototype program and track its execution on Infineon Bot kit platform;

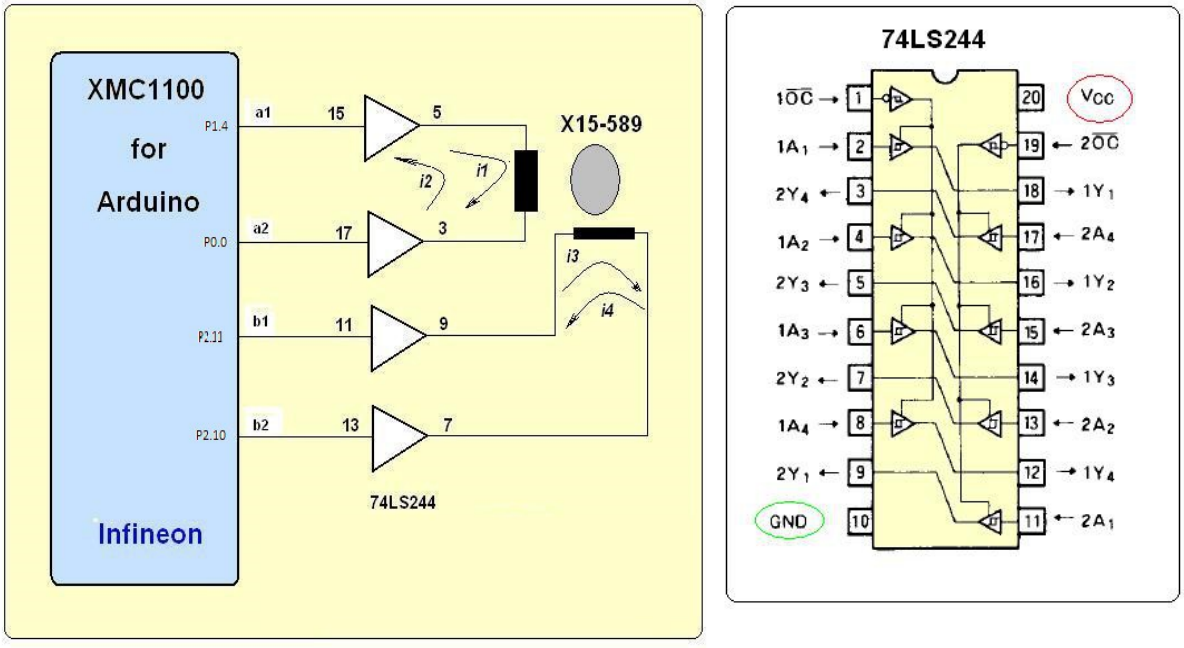

Figure 4.3 The interconnection scheme between micro-motors and XMC

- **E4.6 Problems proposed:**

1. Create a program that will determine the maximum speed of rotation of the motors;
2. Take a sequence program that will allow an initialization of the motor position;
3. Make a program that allows viewing on platform's LED the orders for stepping micro-motor.

Figure 4.4 Infineon XMC1100 LEDs

4. Realize a sequence program that will display an analog signal specified using a potentiometer at one of the AD convertor's input;
5. Make a program that will show on display a numeric value (0-9) received on the Bluetooth interface.

- **E4.7 The experiment can be extended to be used for:**

- Making board instrumentation (Figure 4.5 and 4.6)
- Achieving control panels that require visualization of parameters;
- Achieving control panels for aircraft;
- Industrial instrumentation.

Figure 4.5 The board used in the experiment

Figure 4.6 A commercial board

- **E4.8 More helpful information:**

1. **Board Instrumentation -** https://en.wikipedia.org/wiki/Dashboard
2. **Arduino -XMC** - https://code.google.com/p/arduino-to-xmc/
3. **Sport car display** - http://www.stackltd.com/
4. **Instrumentation class for automobile -** https://en.wikipedia.org/wiki/List_of_vehicle_instruments
5. **Smart Instruments** - http://www.dakotadigital.com/index.cfm/page/ptype=results/category_id=69/mode=cat/cat69.htm
6. **Special step by step motors** - http://www.sonceboz.com/en/slimline-stepper-motor-instrument-cluster/
7. **Arduino XMC Layer -** https://code.google.com/p/arduino-to-xmc/wiki/Arduino_to_XMC_library_layering
8. **Wiring Documentation** - https://code.google.com/p/arduino-to-xmc/wiki/Documentation
9. **IoT Fundamentals for instrumentation** - http://www.lxistandard.org/Documents/LXI%20Primer/The%20LXI%20Primer%201.2b%202015.pdf