FRA 222
Microcontroller Interface

WEEK 8 - DSP

Summary

today topic

DSP – Matrix Calculation

DSP – PID

Debug SWTimeline

DSP – Matrix Calculation

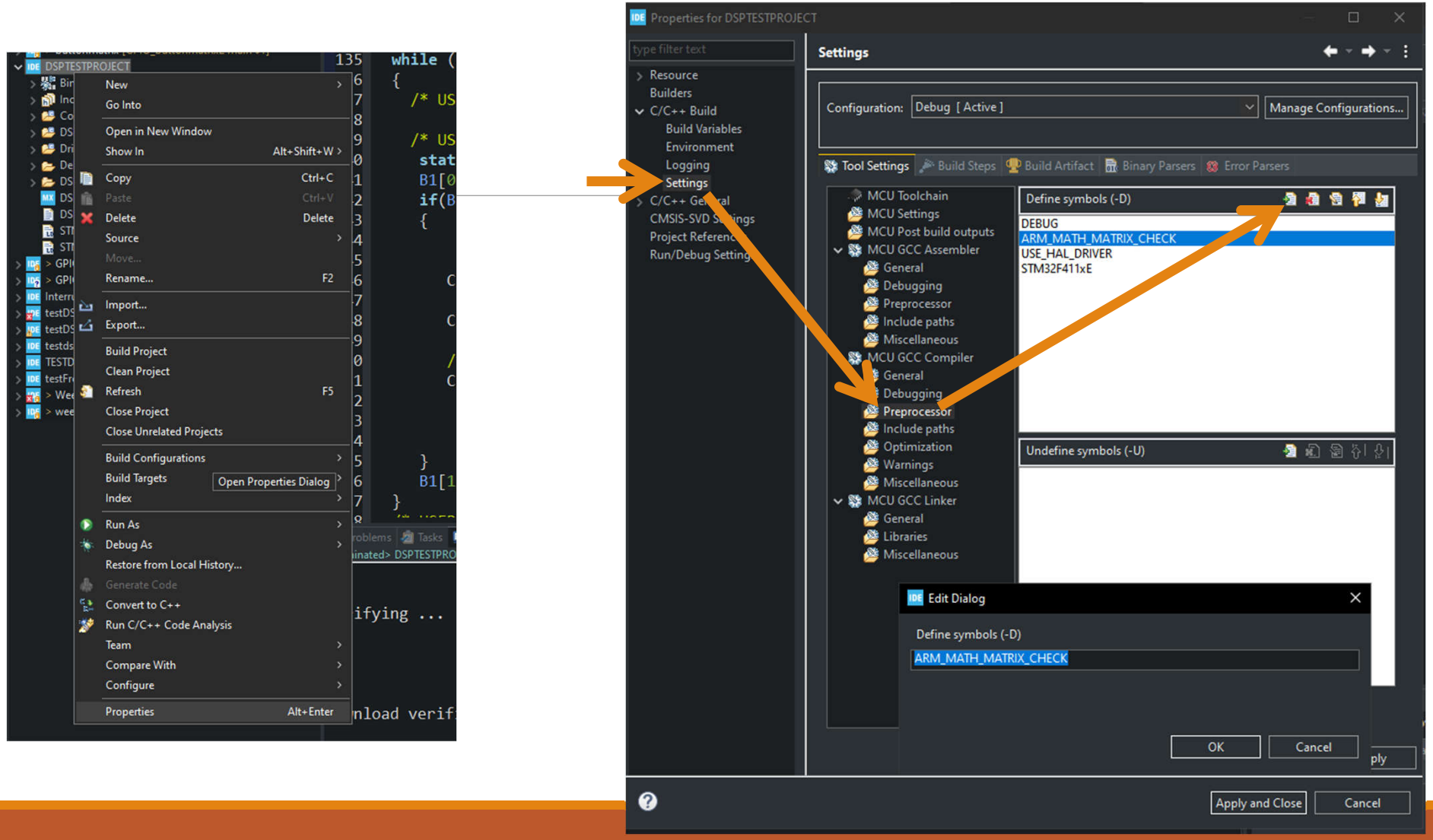
```
/* USER CODE BEGIN Includes */
#include "arm_math.h"

/* USER CODE END Includes */
```

```
/* USER CODE BEGIN PV */

// start Matrix
float32_t A_f32[16] = {
        1,  2,  3,  4,
        5,  6,  7,  8,
        9,  10, 11, 12,
        13, 14, 15, 16
};
arm_matrix_instance_f32 A;

float32_t B_f32[4] = {
        1,
        2,
        3,
        4
};
arm_matrix_instance_f32 B;
```

```
//Output and storage matrix
float32_t At_f32[16];   // A transpose
arm_matrix_instance_f32 At;

float32_t AtmA_f32[16]; // A transpose multiply A  = tran(A) *A
arm_matrix_instance_f32 AtmA;

float32_t AaB_f32[16];  // A Addition B
arm_matrix_instance_f32 AaB;

volatile arm_status CalcSt;
/* USER CODE END PV */
```

```c
/* USER CODE BEGIN 2 */
//init all matrix

arm_mat_init_f32(&A, 4, 4,(float32_t*) &A_f32);
arm_mat_init_f32(&At, 4, 4, (float32_t*) &At_f32);
arm_mat_init_f32(&AtmA, 4, 4, (float32_t*) &AtmA_f32);
arm_mat_init_f32(&B, 4, 1, (float32_t*) &B_f32);
arm_mat_init_f32(&AaB, 4, 4, (float32_t*) &AaB_f32);


/* USER CODE END 2 */
```

```c
/* USER CODE BEGIN 3 */
 static GPIO_PinState B1[2];
 B1[0] = HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin);
 if(B1[0]== 0 && B1[1] == 1)
 {


    CalcSt = arm_mat_trans_f32(&A, &At);

    CalcSt = arm_mat_mult_f32(&At, &A , &AtmA);

    //wrong add
    CalcSt = arm_mat_add_f32(&A, &B, &AaB);



 }
 B1[1] = B1[0];
}
/* USER CODE END 3 */
```


DSPTESTPROJECT
New
Go Into
Open in New Window
Show In
Alt+Shift+W
Copy
Ctrl+C
Paste
Ctrl+V
Delete
Delete
Source
Move...
Rename...
F2
Import...
Export...
Build Project
Clean Project
Refresh
F5
Close Project
Close Unrelated Projects
Build Configurations
Build Targets
Open Properties Dialog
Index
Run As
Debug As
Restore from Local History...
Generate Code
Convert to C++
Run C/C++ Code Analysis
Team
Compare With
Configure
Properties
Alt+Enter
Properties for DSPTESTPROJECT
type filter text
Resource
Builders
C/C++ Build
Build Variables
Environment
Logging
Settings
Settings
Configuration:
Debug [Active]
Manage Configurations...
Tool Settings
Build Steps
Build Artifact
Binary Parsers
Error Parsers
MCU Toolchain
MCU Settings
MCU Post build outputs
MCU GCC Assembler
General
Debugging
Preprocessor
Include paths
Miscellaneous
MCU GCC Compiler
General
Debugging
Preprocessor
Include paths
Optimization
Warnings
Miscellaneous
MCU GCC Linker
General
Libraries
Miscellaneous
Define symbols (-D)
DEBUG
ARM_MATH_MATRIX_CHECK
USE_HAL_DRIVER
STM32F411xE
Undefine symbols (-U)
Edit Dialog
Define symbols (-D)
ARM_MATH_MATRIX_CHECK
OK
Cancel
Apply and Close
Cancel

A Transpose

Input

$$\text{transpose}\left(\begin{pmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\\ 13 & 14 & 15 & 16\end{pmatrix}\right)$$

Output

$$\begin{pmatrix}1 & 5 & 9 & 13\\ 2 & 6 & 10 & 14\\ 3 & 7 & 11 & 15\\ 4 & 8 & 12 & 16\end{pmatrix}$$

A Transpose * A

Input

$$\left(\begin{pmatrix}1 & 5 & 9 & 13\\ 2 & 6 & 10 & 14\\ 3 & 7 & 11 & 15\\ 4 & 8 & 12 & 16\end{pmatrix}\right)\left(\begin{pmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\\ 13 & 14 & 15 & 16\end{pmatrix}\right)$$

Solution steps

Output

$$\begin{pmatrix}276 & 304 & 332 & 360\\ 304 & 336 & 368 & 400\\ 332 & 368 & 404 & 440\\ 360 & 400 & 440 & 480\end{pmatrix}$$

A*B

Input

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Solution steps

Matrix multiplication is defined if the number of columns of the first matrix is equal to the number of rows of the

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Multiply each element of the first row of the first matrix by the corresponding element of the first column of the
these products to obtain the element in the first row, first column of the product matrix.

$$\begin{pmatrix} 1+2\cdot 2+3\cdot 3+4\cdot 4 \\ \{\} \\ \{\} \\ \{\} \end{pmatrix}$$

The remaining elements of the product matrix are found in the same way.

$$\begin{pmatrix} 1+2\cdot 2+3\cdot 3+4\cdot 4 \\ 5+6\cdot 2+7\cdot 3+8\cdot 4 \\ 9+10\cdot 2+11\cdot 3+12\cdot 4 \\ 13+14\cdot 2+15\cdot 3+16\cdot 4 \end{pmatrix}$$

Simplify each element by multiplying the individual terms.

$$\begin{pmatrix} 1+4+9+16 \\ 5+12+21+32 \\ 9+20+33+48 \\ 13+28+45+64 \end{pmatrix}$$

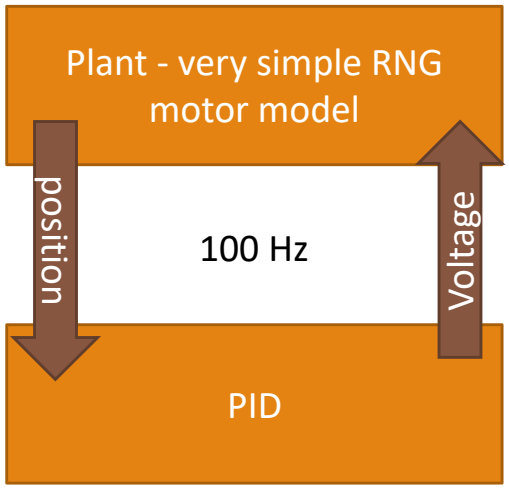
Sum each element of the matrix.

$$\begin{pmatrix} 30 \\ 70 \\ 110 \\ 150 \end{pmatrix}$$

Output

$$\begin{pmatrix} 30 \\ 70 \\ 110 \\ 150 \end{pmatrix}$$

PID controller




```
arm_pid_instance_f32 PID = {0};
float position =0;
float setposition =0;
float Vfeedback = 0;
/* USER CODE END PV */
```

```
/* USER CODE BEGIN PFP */
float PlantSimulation(float VIn) ;

/* USER CODE END PFP */
```

```c
/* USER CODE BEGIN 4 */

float PlantSimulation(float VIn) // run with fix frequency
{

    static float speed =0;
    static float position =0;
    float current= VIn - speed *  0.0123;
    float torque = current * 0.456;
    float acc = torque * 0.789;
    speed += acc;
    position += speed;
    return position;

}
/* USER CODE END 4 */
```

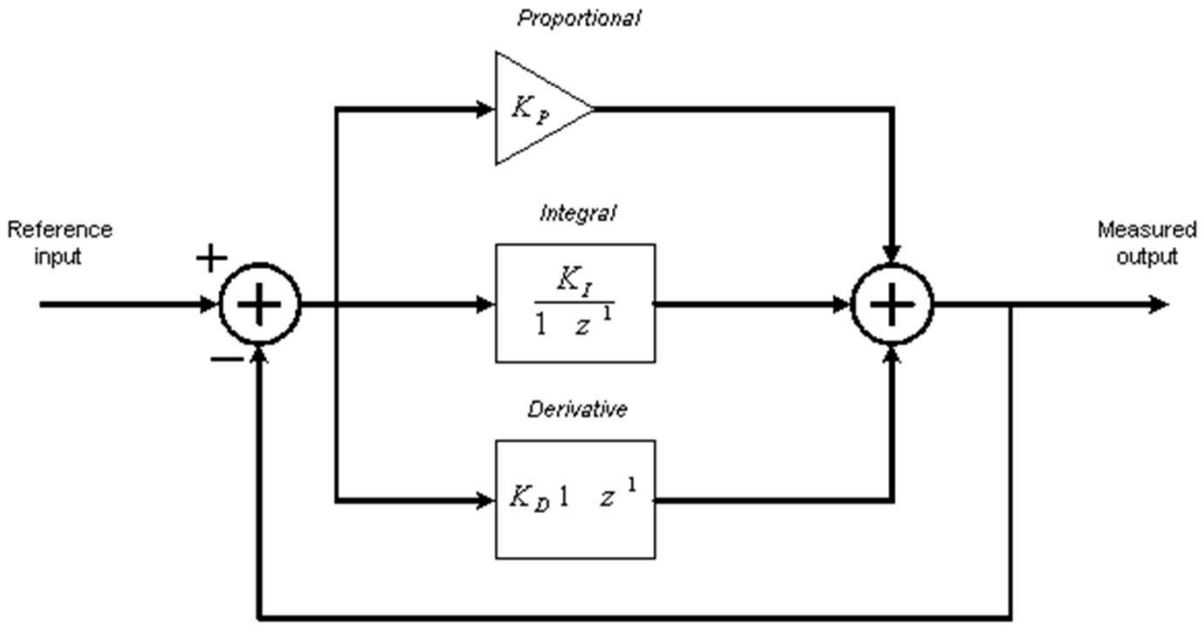
```c
/* USER CODE BEGIN 2 */
PID.Kp =0.1;
PID.Ki =0.00001;
PID.Kd = 0.1;
arm_pid_init_f32(&PID, 0);
```

```c
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    static uint32_t timestamp =0;
    if(timestamp < HAL_GetTick())
    {
        timestamp = HAL_GetTick()+10;

        Vfeedback = arm_pid_f32(&PID, setposition - position);
        position = PlantSimulation(Vfeedback);

    }

  /* USER CODE END WHILE */
```

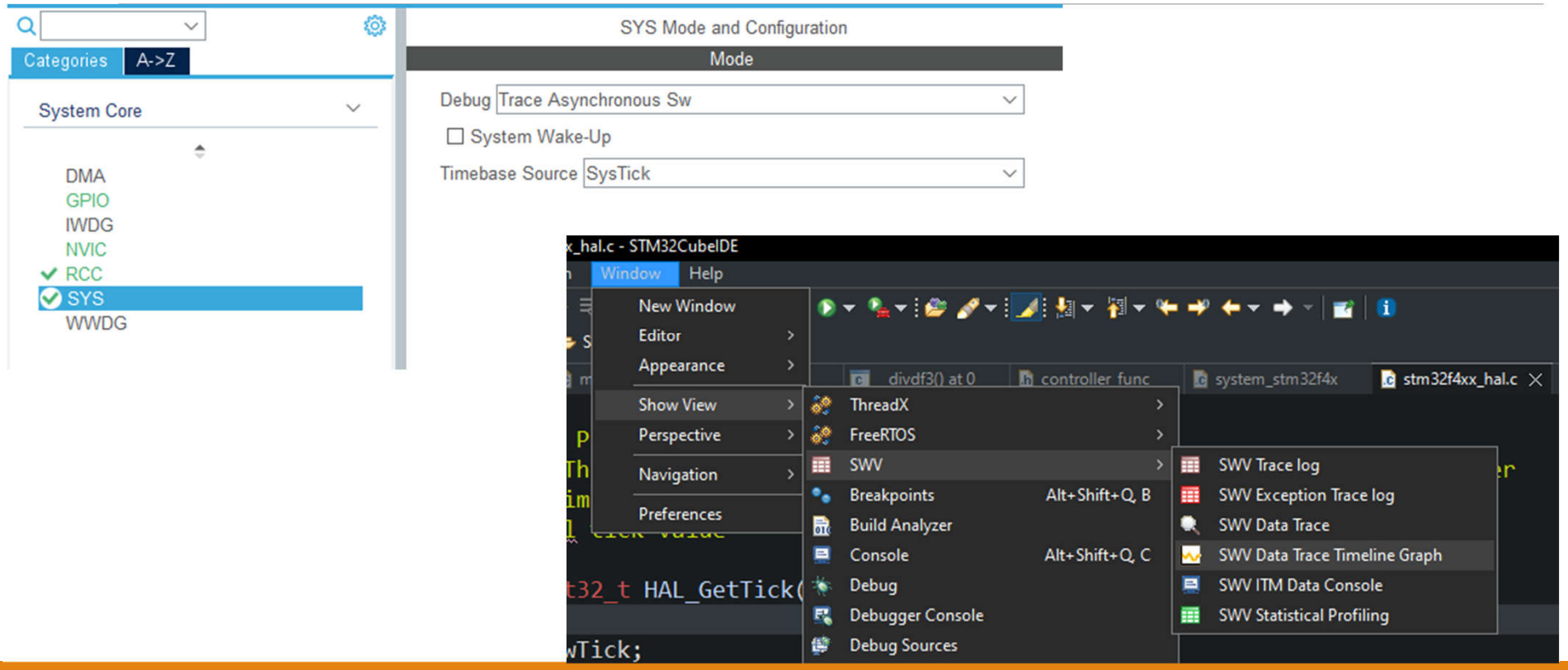
Algorithm:

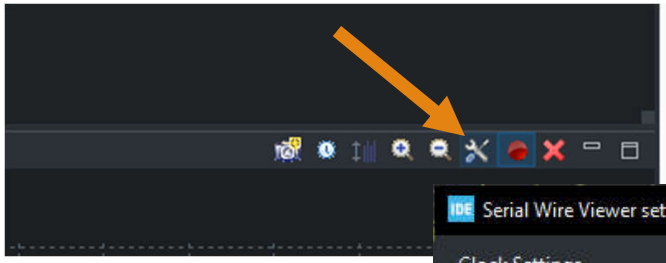
```
y[n] = y[n-1] + A0 * x[n] + A1 * x[n-1] + A2 * x[n-2]
A0 = Kp + Ki + Kd
A1 = (-Kp ) - (2 * Kd )
A2 = Kd
```

where `Kp` is proportional constant, `Ki` is Integral constant and `Kd` is Derivative constant



Debug SWTimeline





Serial Wire Viewer settings forDSPTESTPROJECT Debug

Clock Settings

Core Clock: 84 MHz

Clock Prescaler: 42

SWO Clock: 2000.0 kHz

Trace Events

- CPI: Cycles per instruction
- EXC: Exception overhead
- SLEEP: Sleep cycles
- LSU: Load store unit cycles
- FOLD: Folded instructions
- EXETRC: Trace Exceptions

PC Sampling

Enable Resolution: 1024 Cycles/sample

Timestamps

Enable Prescaler: 1

Data Trace

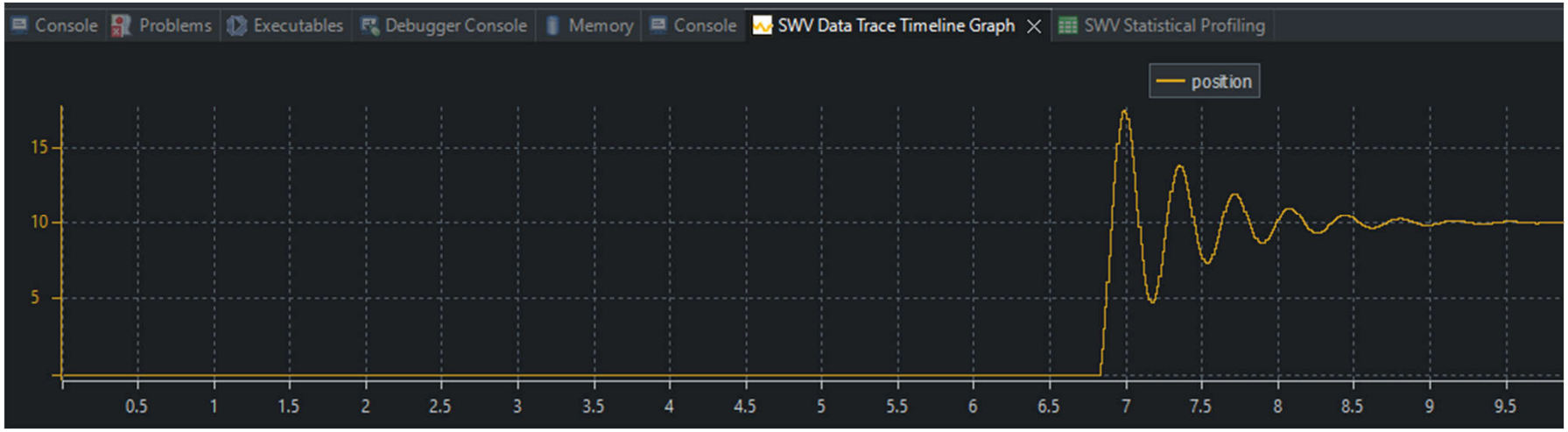
	Comparator 0	Comparator 1	Comparator 2	Comparator 3
Enable	☑	☐	☐	☐
Var/Addr:	position	0x0	0x0	0x0
Access:	Write	Read/Write	Read/Write	Read/Write
Size:	Word	Word	Word	Word
Generate:	Data Value	Data Value	Data Value	Data Value

ITM Stimulus Ports

Enable port: 31 ■■■■■■■■ 24 23 ■■■■■■■■ 16 15 ■■■■■■■■ 8 7 ■■■■■■■■ 0

Privileged only ports: ☐ Port 31..24 ☐ Port 23..16 ☐ Port 15..8 ☐ Port 7..0

OK Cancel


Console
Problems
Executables
Debugger Console
Memory
Console
SWV Data Trace Timeline Graph
SWV Statistical Profiling
position
15
10
5
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8
8.5
9
9.5

LAB 4 Motor position control With PID

due date : 19-20 April 2023

เขียนระบบควบคุมโดยใช้ความรู้จาก วิชา control มาช่วยในการสร้างระบบควบคุมตำแหน่งอย่างง่าย

โดยกำหนดให้

-Error : with in 1 degree form setpoint

-rotation Range : 0 – 36000 degree

-ระบบต้องหยุดนิ่งภายใน 1 วินาที หลังจากถึงจุด setpoint

-ระบบจะต้องพยายามเข้าสู่ setpoint **ให้เร็วที่สุด**