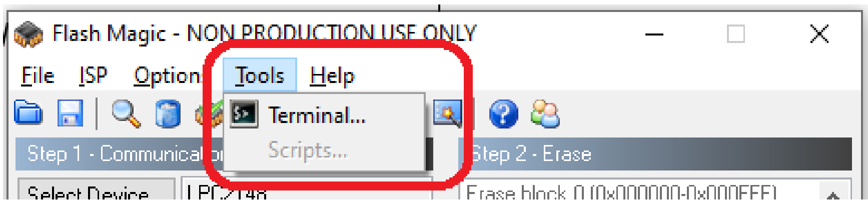
HARDWARE WORKING PROGRAMS

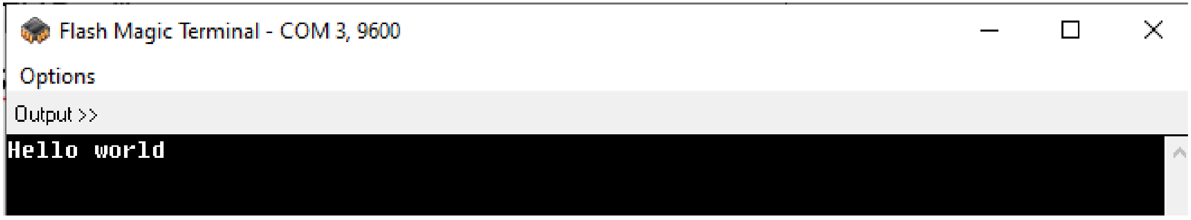
9b. Display “Hello World” message using Internal UART.	Decoding part

```
#include<LPC214X.H>
char *msg = "Hello world  ";
int main()
{
PINSEL0 = 0X5;
U0LCR = 0X83;
U0DLM= 0X00;
U0DLL = 0X61;
U0LCR = 0X03;
while(*msg!= 0X00)
{
while (!(U0LSR & 0X20));
U0THR =*msg;
 msg++;
}
}
```

OUTPUT: **After dumping the Hex file in ARM kit via Flash Magic >>>>> go to Tools click on >>>>> Terminal**

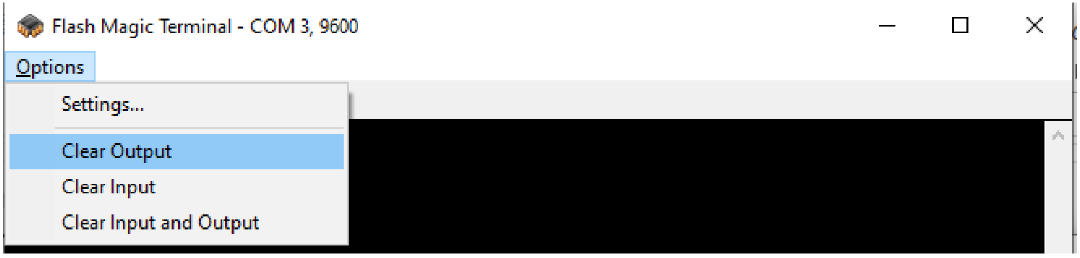


(Check for Port # & Baud Rate: 9600) Click on OK “Hello World” message will display.



```
Hello world
```

How to Clear the Message: Click on Options >>>> Clear Output



10b. Interface & Control a DC Motor	Decoding part

```
#include<LPC214x.H>
#define CLK1 (1<<16)
#define ACLK2 (1<<17)
#define STOP3 (1<<18)
int main (void)
{
//IO1DIR=0X0007000;
IODIR0 =0XE0;
IOSET0 = 0X80;
while(1)
{
if (!(IOPIN1 &CLK1))
{
IOSET0= (1<<5); IOCLR0=(1<<6);
}
if (!(IOPIN1 & ACLK2))
{
IOCLR0= (1<<5); IOSET0=(1<<6);
}
if (!(IOPIN1 & STOP3))
{
IOCLR0= (1<<5); IOCLR0=(1<<6);
}
}
}
```

OUTPUT: Connect DC MOTOR to J2 on the ARM kit After dumping the Hex file in ARM kit via Flash Magic >>>>> on the ARM kit concentrate S8 – S1 Keys / out of 8-Keys only 3-Keys are assigned for DC MOTOR control.

Press S1: for CLOCLWISE ROTATION
Press S2: for ANTI-CLOCLWISE ROTATION
Press S3: to STOP the DC-Motor

11b. Interface a Stepper motor & rotate it in clockwise and anti-clockwise direction.

Decoding part

```
#include <LPC214x.H>
#define CLK (1 << 16)
//KEY1, P1.16
#define ACLK (1 << 17)
//KEY5, P1.17
void delay(unsigned int
count)
{
  int j=0,i=0;
  for(j=0;j<count;j++)
  {
    for(i=0;i<500;i++);
  }
}
int main (void)
{
int i;
IO1DIR = 0x0F000000;
IO1CLR = 0x0F000000;
PINSEL2 = 0x0;
while(1)
{
if (!(IO1PIN & ACLK) )
for(i=0;i<50;i++)
{
IO1PIN = 0X01000000;
delay(200);
IO1PIN =
0X02000000;
delay(200);
IO1PIN =
0X04000000;
delay(200);
IO1PIN =
0X08000000;
delay(200);
}
if (!(IO1PIN & CLK) )
for(i=0;i<50;i++)
{
IO1PIN =
0X08000000;
delay(200);
IO1PIN =
0X04000000;
delay(200);
IO1PIN =
0X02000000;
delay(200);
IO1PIN =
0X01000000;
delay(200);
}
}
}
```

OUTPUT: Connect STEPPER-Motor to J3 on the ARM kit, Provide supply from External Power Adopter (12v, 1.0A) Bigger Diameter should be connected to J5 in the POWER SUPPLY portion of the kit and make toggle switch-SW1 ON to rotate the STEPPER-Motor.

After dumping the Hex file in ARM kit via Flash Magic >>>>> concentrate on S8 – S1 Keys / out of 8-Keys only 2-Keys are assigned for STEPPER-Motor control.

Press S1: for CLOCLWISE STEP ROTATION
Press S2: for ANTI-CLOCLWISE STEP ROTATION

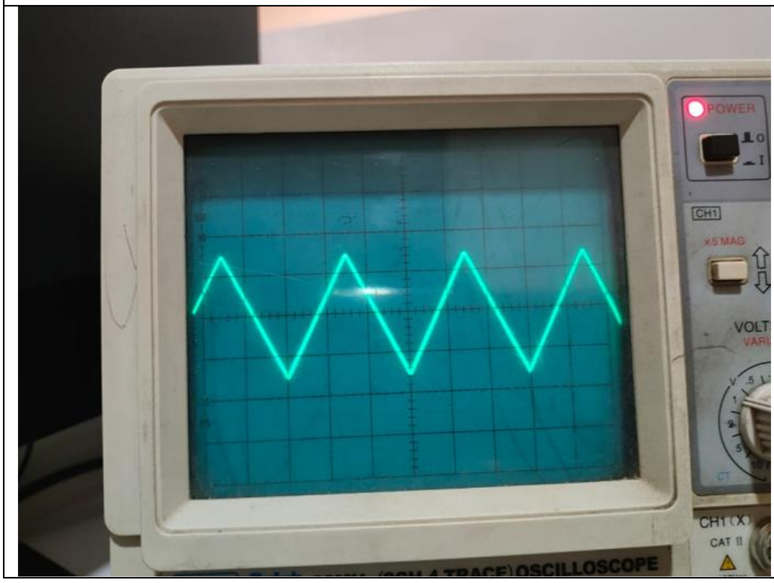
13b. Interface a DAC and generate Triangular and Square waveforms

Decoding part

TRIANGULAR WAVE FORM

```
#include <LPC214x.H>
Init_DAC()
{
PINSEL1 = 0X00080000;
DACR = 0;
}
Write_DAC(unsigned int dacval)
{
DACR = dacval<< 6;
}
int main (void)
{
unsigned int i; Init_DAC();
while(1)
{
for(i=0;i<1024;i++)
Write_DAC(i);
for(i=1024;i>0;i--)
Write_DAC(i);
}
}
```

OUTPUT: CRO is required to view the waveforms use either CH1 / CH2 (CHANNEL 1 / 2)
CRO Probe ends with RED colour is +ve should connect to P2 &
BLACK colour is -ve should connect to P3 on the ARM kit (below POWER SUPPLY UNIT portion).
After dumping the Hex file in ARM kit via Flash Magic >>>>> Observe TRIANGULAR wave on the CRO screen.



Check the continuity of the CRO-Probe for its working.

CRO Settings:

Switch ON the Power Button of CRO (Black in Colour) CH1 / CH2 to be pressed for which CRO-Probe is connected rest other buttons should be in release position, set VOLTS/DIV knob to 1 or 0.5V, set TIME/DIV knob to .5 or .2 ms.

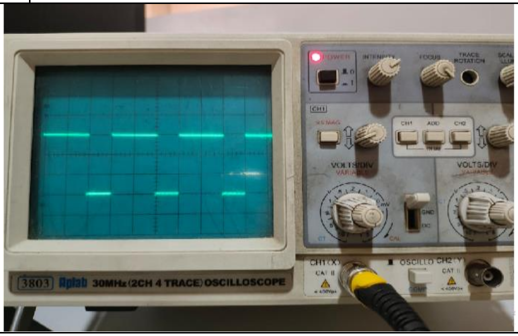
13b. Interface a DAC and generate Triangular and Square waveforms

Decoding part

SQUARE WAVE FORM

```
#include <LPC214x.H> /*
LPC214x definitions */
Init_DAC()
{
PINSEL1 = 0X00080000;
DACR = 0;
}
Write_DAC(unsigned int
dacval)
{
DACR = dacval<< 6;
}
void delay(unsigned int
count)
{
int j=0,i=0;
for(j=0;j<count;j++)
{
for(i=0;i<120;i++);
}
}
int main (void)
{
Init_DAC(); while(1)
{
Write_DAC(00);
delay(10); // change this
value to change Frequency =
TON
Write_DAC(1000); // change
this value to change
Amplitude delay(100);
delay(20); // change
this value to change
Frequency = TOFF
}
}
{
IO1PIN = 0X01000000;
delay(200);
IO1PIN = 0X02000000;
delay(200);
IO1PIN = 0X04000000;
delay(200);
IO1PIN = 0X08000000;
delay(200);
}
if (!(IO1PIN & CLK) )
for(i=0;i<50;i++)
{
IO1PIN = 0X08000000;
delay(200);
IO1PIN = 0X04000000;
delay(200);
IO1PIN = 0X02000000;
delay(200);
IO1PIN = 0X01000000;
delay(200);
}
}
}
```

OUTPUT: CRO is required to view the waveforms use either CH1 / CH2 (CHANNEL 1 / 2) CRO Probe ends with RED colour is +ve should connect to P2 & BLACK colour is -ve should connect to P3 on the ARM kit (below POWER SUPPLY UNIT). After dumping the Hex file in ARM kit via Flash Magic >>>>> Observe SQUARE wave on the CRO screen.



13b. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between	Decoding part

ALPHA-NUMERIC CHARACTERS

(0 1 2 3 4 5 6 7 8 9 A b C d E F)

```
#include<LPC214x.H>
#define DIG1 (1<<10)
unsigned char
ssc[]={0x88,0xeb,0x4c,0x49,0x2b,0x19,0x18,
0xcb,0x8,0x9,0xa,0x38,0x9c,0x68,0x1c,0x1e}
;
void delay()
{
unsigned int i,j=0;
for( i=0;i<4000000;i++) j++;
}
int main(void)
{
int count;
IODIR0 = 0XFFFFFFFF;
while(1)
{
for( count=0;count<16;count++)
{
IOCLR0 = 0x007F8000;
IOSET0= (ssc[count]<<15);
IOSET0 = DIG1;
delay();
}
}
}
```

OUTPUT: Four 7-Segment Display units D12, D11, D10 & D9 are on the ARM kit Concentrate on the 7-Segment Display unit D9 After dumping the Hex file in ARM kit via Flash Magic >>>>> Observe

ALPHA-NUMERIC CHARACTERS

(0 1 2 3 4 5 6 7 8 9 A b C d E F) on the D9.