EMBEDDED SYSTEMS
LAB. REPORT VII

HACETTEPE UNIVERSITY

COMPUTER SCIENCE
DEPARTMENT

TEAM NAME:

KENDINI MARUL SANAN MARUL

MEMBERS:

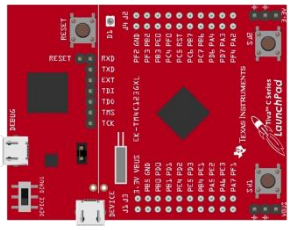
KUTAY BARCIN 21526715

DEFNE TUNCER 21627686

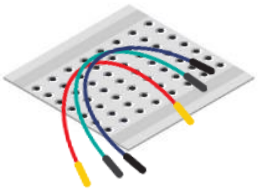
ADC & DAC

Eight different musical notes are generated using DAC. Potentiometer is used to switch between notes.

STEP1: EQUIPMENT

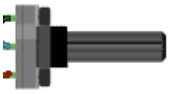


TM4C123G:



BREADBOARD AND WIRES:

ADC



10K POTENTIOMETER:

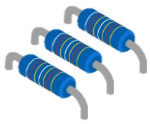
DAC

HEADPHONE JACK:

3 or 5 pin stereo jack

RESISTORS:

Three 1.5K resistors



Three 12K resistors

STEP2: PLL

PLL initialization is done:

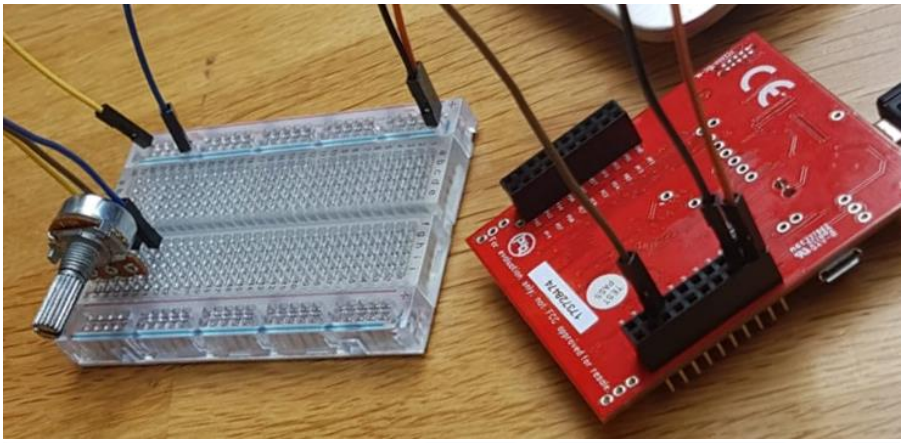
```
void PLL_Init(void){
  // 0) Use RCC2
  SYSCTL_RCC2_R |=  0x80000000;  // USERCC2
  // 1) bypass PLL while initializing
  SYSCTL_RCC2_R |=  0x00000800;  // BYPASS2, PLL bypass
  // 2) select the crystal value and oscillator source
  SYSCTL_RCC_R = (SYSCTL_RCC_R &~0x000007C0)   // clear XTAL field, bits 10-6
                 + 0x00000540;   // 10101, configure for 16 MHz crystal
  SYSCTL_RCC2_R &= ~0x00000070;  // configure for main oscillator source
  // 3) activate PLL by clearing PWRDN
  SYSCTL_RCC2_R &= ~0x00002000;
  // 4) set the desired system divider
  SYSCTL_RCC2_R |= 0x40000000;   // use 400 MHz PLL
  SYSCTL_RCC2_R = (SYSCTL_RCC2_R&~ 0x1FC00000)  // clear system clock divider
                  + (4<<22);      // configure for 80 MHz clock
  // 5) wait for the PLL to lock by polling PLLLRIS
  while((SYSCTL_RIS_R&0x00000040)==0){};  // wait for PLLRIS bit
  // 6) enable use of PLL by clearing BYPASS
  SYSCTL_RCC2_R &= ~0x00000800;
}
```

STEP3: ADC

PORTE initialization is done: Channel is set to 1. Analog voltage value is read from PE2 which changes between 0 to 4095 with turning of the pot. **ADC0_InSeq3** is used to read the result.

```
void ADC0_InitSWTriggerSeq3_Ch1(void){ volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x00000010;   // 1) activate clock for Port E
  delay = SYSCTL_RCGC2_R;         //    allow time for clock to stabilize
  GPIO_PORTE_DIR_R &= ~0x04;      // 2) make PE2 input
  GPIO_PORTE_AFSEL_R |= 0x04;     // 3) enable alternate function on PE2
  GPIO_PORTE_DEN_R &= ~0x04;      // 4) disable digital I/O on PE2
  GPIO_PORTE_AMSEL_R |= 0x04;     // 5) enable analog function on PE2
  SYSCTL_RCGC0_R |= 0x00010000;   // 6) activate ADC0
  delay = SYSCTL_RCGC2_R;
  SYSCTL_RCGC0_R &= ~0x00000300; // 7) configure for 125K
  ADC0_SSPRI_R = 0x0123;          // 8) Sequencer 3 is highest priority
  ADC0_ACTSS_R &= ~0x0008;        // 9) disable sample sequencer 3
  ADC0_EMUX_R &= ~0xF000;         // 10) seq3 is software trigger
  ADC0_SSMUX3_R &= ~0x000F;       // 11) clear SS3 field
  ADC0_SSMUX3_R += 1;             //    set channel Ain1 (PE2)
  ADC0_SSCTL3_R = 0x0006;         // 12) no TS0 D0, yes IE0 END0
  ADC0_ACTSS_R |= 0x0008;         // 13) enable sample sequencer 3
}

unsigned long ADC0_InSeq3(void){  unsigned long result;
  ADC0_PSSI_R = 0x0008;            // 1) initiate SS3
  while((ADC0_RIS_R&0x08)==0){};   // 2) wait for conversion done
  result = ADC0_SSFIFO3_R&0xFFF;   // 3) read result
  ADC0_ISC_R = 0x0008;             // 4) acknowledge completion
  return result;
}
```

When the pot is fully turned at 270-degree angle, analog voltage value is read 4095.



Watch 1

Name	Value	Type
ADCvalue	4095	unsigned long

STEP4: 4-BIT BINARY-WEIGHTED DAC

If a sequence of numbers that form a sine wave is outputted to the DAC then a continuous tone can be heard from a speaker/headphone/buzzer. Since we will be using 4-bit binary weighted DAC, outputs vary from 0 to 15. The **pitch** is defined as the **frequency** of the wave.

```
const unsigned char SineWave[32] = {8,9,11,12,13,14,14,15,15,15,14,14,13,12,11,9,8,7,5,4,3,2,2,1,1,1,2,2,3,4,5,7};
```

Notes		Frequency	SysTick interrupts 32 times per period	RELOAD value (80MHz clock)
C	Do	262Hz	32*262Hz = 8384Hz	9542
D	Re	294Hz	32*294Hz = 9408Hz	8503
E	Mi	330Hz	32*330Hz = 10560Hz	7576
F	Fa	349Hz	32*349Hz = 11168Hz	7163
G	Sol	393Hz	32*393Hz = 12576Hz	6361
A	La	440Hz	32*440Hz = 14080Hz	5682
B	Si	494Hz	32*494Hz = 15808Hz	5061
C	Do	523Hz	32*523Hz = 16736Hz	4780

Table 2 contains frequency and RELOAD values for the notes in an octave.

N	PB3	PB2	PB1	PB0	Vout (V)
0	0	0	0	0	3.3V * 0/15 = 0V
1	0	0	0	1	3.3V * 1/15 = 0.22V
2	0	0	1	0	3.3V * 2/15 = 0.44V
3	0	0	1	1	3.3V * 3/15 = 0.66V
4	0	1	0	0	3.3V * 4/15 = 0.88V
5	0	1	0	1	3.3V * 5/15 = 1.1V
6	0	1	1	0	3.3V * 6/15 = 1.32V
7	0	1	1	1	3.3V * 7/15 = 1.54V
8	1	0	0	0	3.3V * 8/15 = 1.76V
9	1	0	0	1	3.3V * 9/15 = 1.98V
10	1	0	1	0	3.3V * 10/15 = 2.2V
11	1	0	1	1	3.3V * 11/15 = 2.42V
12	1	1	0	0	3.3V * 12/15 = 2.64V
13	1	1	0	1	3.3V * 13/15 = 2.86V
14	1	1	1	0	3.3V * 14/15 = 3.08V
15	1	1	1	1	3.3V * 15/15 = 3.3V

Table 1 Specifications of then 4-bit binary-weighted DAC.

PB3 1.5k resistor

PB2 3k resistor (2 serial 1.5k resistors)

PB1 6k resistor (2 parallel 12k resistors)

PB0 12k resistor

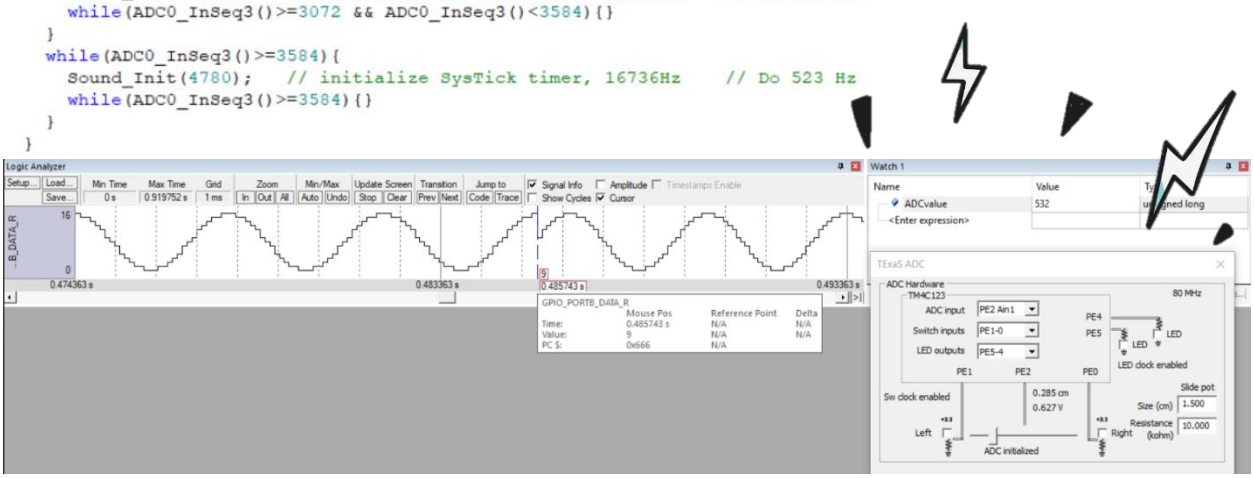
PORTB initialization is done: SysTick interrupts generate Sine wave.

```
void DAC_Init(void){unsigned long volatile delay;
  SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOB; // activate port B
  delay = SYSCTL_RCGC2_R;    // allow time to finish activating
  GPIO_PORTB_AMSEL_R &= ~0x0F;      // no analog
  GPIO_PORTB_PCTL_R &= ~0x0000FFFF; // regular GPIO function
  GPIO_PORTB_DIR_R |= 0x0F;       // make PB3-0 out
  GPIO_PORTB_AFSEL_R &= ~0x0F;    // disable alt funct on PB3-0
  GPIO_PORTB_DEN_R |= 0x0F;       // enable digital I/O on PB3-0
}

void Sound_Init(unsigned long period){
  DAC_Init();          // Port B is DAC
  Index = 0;
  NVIC_ST_CTRL_R = 0;         // disable SysTick during setup
  NVIC_ST_RELOAD_R = period-1;// reload value
  NVIC_ST_CURRENT_R = 0;      // any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x20000000; // priority 1
  NVIC_ST_CTRL_R = 0x0007; // enable SysTick with core clock and interrupts
}

void SysTick_Handler(void){
  Index = (Index+1)&0x1F;                 // 8,9,11,12,13,14,14,15,15,15,14,...
  GPIO_PORTB_DATA_R = SineWave[Index];    // output one value each interrupt
}
```

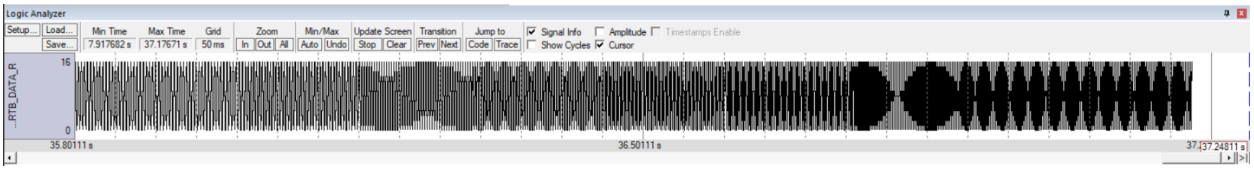
STEP5: MAIN

As pot turns, read voltage value changes, and starting from 4095, with each 512 unit change a different note's sine wave is initialized.

```
int main(void){unsigned long volatile delay;
  PLL_Init();          // bus clock at 80 MHz
  ADC0_InitSWTriggerSeq3_Ch1();
  while(1){
    while(ADC0_InSeq3()<512){
      Sound_Init(9542);   // initialize SysTick timer, 8384Hz     // Do 262 Hz
      while(ADC0_InSeq3()<512){}
    }
    while(ADC0_InSeq3()>=512 && ADC0_InSeq3()<1024){
      Sound_Init(8503);   // initialize SysTick timer, 9408Hz     // Re 294 Hz
      while(ADC0_InSeq3()>=512 && ADC0_InSeq3()<1024){}
    }
    while(ADC0_InSeq3()>=1024 && ADC0_InSeq3()<1536){
      Sound_Init(7576);   // initialize SysTick timer, 10560Hz    // Mi 330 Hz
      while(ADC0_InSeq3()>=1024 && ADC0_InSeq3()<1536){}
    }
    while(ADC0_InSeq3()>=1536 && ADC0_InSeq3()<2048){
      Sound_Init(7163);   // initialize SysTick timer, 11168Hz    // Fa 349 Hz
      while(ADC0_InSeq3()>=1536 && ADC0_InSeq3()<2048){}
    }
    while(ADC0_InSeq3()>=2048 && ADC0_InSeq3()<2560){
      Sound_Init(6361);   // initialize SysTick timer, 12576Hz    // Sol 393 Hz
      while(ADC0_InSeq3()>=2048 && ADC0_InSeq3()<2560){}
    }
    while(ADC0_InSeq3()>=2560 && ADC0_InSeq3()<3072){
      Sound_Init(5682);   // initialize SysTick timer, 14080Hz    // La 440 Hz
      while(ADC0_InSeq3()>=2560 && ADC0_InSeq3()<3072){}
    }
    while(ADC0_InSeq3()>=3072 && ADC0_InSeq3()<3584){
      Sound_Init(5061);   // initialize SysTick timer, 15808Hz    // Si 494 Hz
      while(ADC0_InSeq3()>=3072 && ADC0_InSeq3()<3584){}
    }
    while(ADC0_InSeq3()>=3584){
      Sound_Init(4780);   // initialize SysTick timer, 16736Hz    // Do 523 Hz
      while(ADC0_InSeq3()>=3584){}
    }
  }
```



AS POT TURNS FROM 0-DEGREE ANGLE TO 34-DEGREE ANGLE (ABOVE 512) GENERATED NOTE CHANGES FROM DO TO RE.



DO RE MI FA SOL LA SI DO

STEP7: CIRCUIT DESIGN