This semester we have decided to standardize the I/O Pin assignment to make remote debugging a little bit easier for the TAs. The standardized Signal and Port assignments are listed below:

PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-A[2]	ST7735_SCK	SSI0_SCK	SSI interface to the ST7735 LCD
Port-A[3]	ST7735_TFT_CS	SSI0_SS	
Port-A[4]	ST7735_MISO	SSI0_MISO	
Port-A[5]	ST7735_MOSI	SSI0_MOSI	
Port-A[6]	ST7735_DC	GPIO	
Port-A[7]	ST7735_RST	GPIO	

PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-B[0]	ST7735_CARD_CS	GPIO	ST7735 SDCARD Chip Select
Port-B[1]	PB1	GPIO	Used for simple speaker
Port-B[2]	PB2	GPIO/I2C0_SCK	Primarily GPIO, can be used for I2C
Port-B[3]	PB3	GPIO/I2C0_SDA	
Port-B[4]	TACH2	T1_CCP0	Tach Input for Lab-10
Port-B[5]	PB5	GPIO/AIN_11	GPIO or Analog input
Port-B[6]	PWM	M0_PWM0	PWM control for Lab-10
Port-B[7]	PB7	GPIO/PWM1	

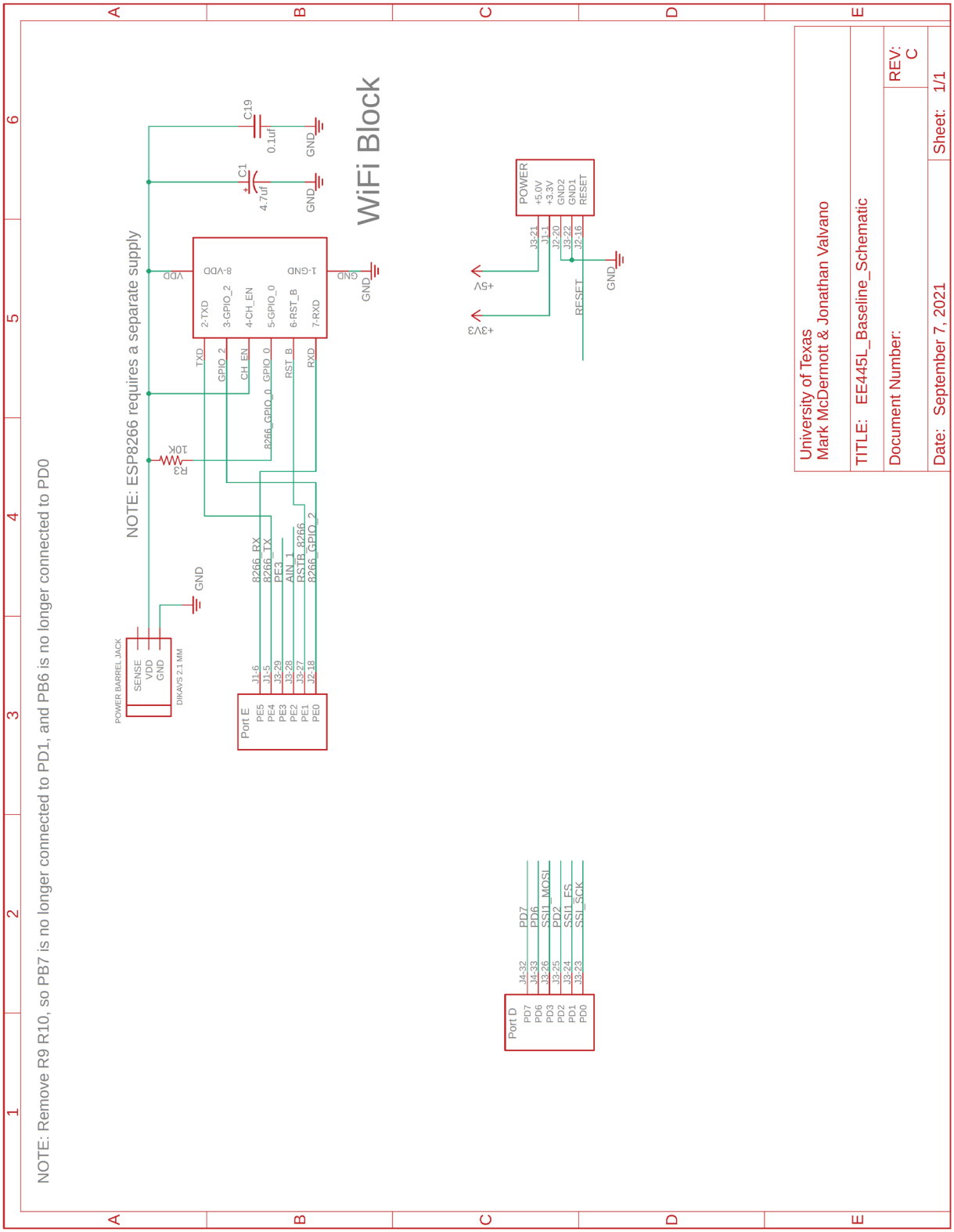
PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-C[4]	UP	GPIO	Used for switches in Lab 3
Port-C[5]	RIGHT	GPIO	
Port-C[6]	LEFT	GPIO	
Port-C[7]	DOWN	GPIO	

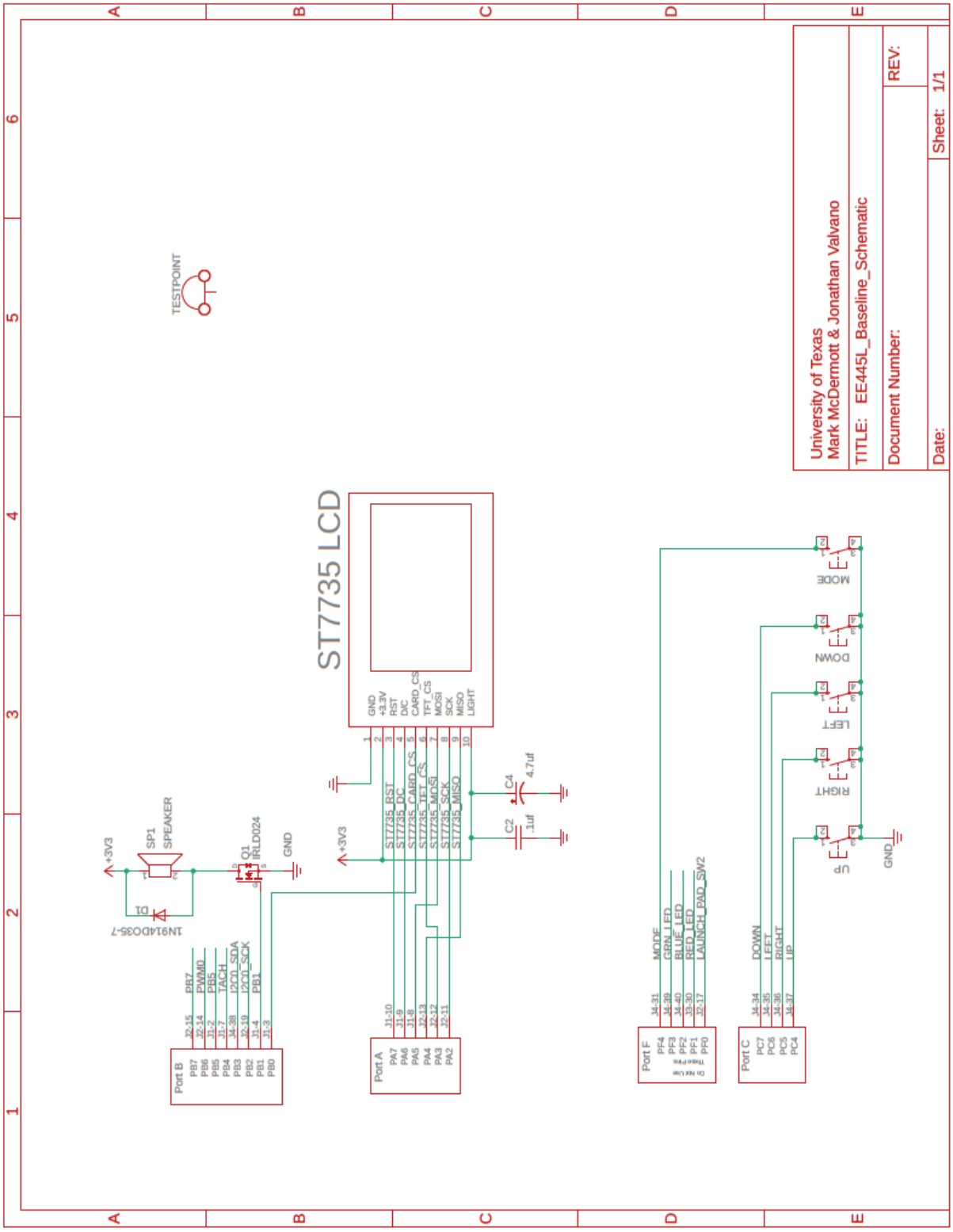
PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-D[0]	SSI1_SCK	SSI1_SCK	SCK to TLV5616 DAC
Port-D[1]	SSI1_FS	SSI1_FS	SS/FS to TLV5616 DAC
Port-D[2]	PD2	AIN_5	Used to monitor DAC_OUT
Port-D[3]	SSI1_MOSI	SSI1_MOSI	MOSI to TLV5616 DAC
Port-D[6]	PD6	GPIO	Can be used as alternate UART Port
Port-D[7]	PD7	GPIO	

PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-E[0]	8266_GPIO_2	GPIO	Status signal 8266 WiFi
Port-E[1]	RSTB_8266	GPIO	Reset signal to 8266 WiFi
Port-E[2]	AIN_1	AIN1	Used for Audio Input Lab-9
Port-E[3]	GPIO	GPIO	
Port-E[4]	8266_TX	U5_RX	UART TXD/RXD to 8266 WiFi
Port-E[5]	8266_RX	U5_TX	

PORT	SIGNAL NAME	TM4C PORT ASSIGNMENT	NOTES
Port-F[0]	LAUNCHPAD_SW2	RESERVED for Launchpad	Mode Switch
Port-F[1]	RED_LED		Red LED on LaunchPad
Port-F[2]	BLUE_LED		Blue LED on LaunchPad
Port-F[3]	GREEN_LED		Green LED on LaunchPad
Port-F[4]	LAUNCHPAD_SW1		Mode Switch

The baseline schematic printout is shown on the next two pages. Use this as the starting point for Eagle Schematics

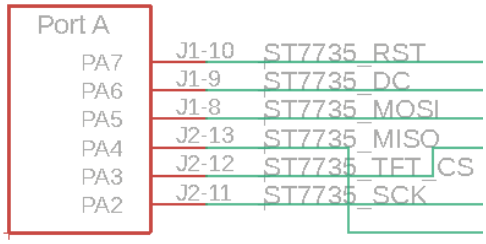

WiFi Block
NOTE: ESP8266 requires a separate supply
NOTE: Remove R9 R10, so PB7 is no longer connected to PD1, and PB6 is no longer connected to PD0
C19
0.1uf
C1
4.7uf
GND
VDD
8-VDD
1-GND
2-TXD
3-GPIO_2
4-CH_EN
5-GPIO_0
6-RST_B
7-RXD
TXD
GPIO_2
CH_EN
GPIO_0
RST_B
RXD
8266_GPIO_0
R3
10K
8266_RX
8266_TX
PE3
AIN_1
RSTB_8266
8266_GPIO_2
Port E
PE5
PE4
PE3
PE2
PE1
PE0
J1-6
J1-5
J3-29
J3-28
J3-27
J2-18
POWER BARREL JACK
SENSE
VDD
GND
DIKAVS 2.1 MM
POWER
+5.0V
+3.3V
GND2
GND1
RESET
J3-21
J1-1
J2-20
J3-22
J2-16
+5V
+3V3
Port D
PD7
PD6
PD3
PD2
PD1
PD0
J4-32
J4-33
J3-26
J3-25
J3-24
J3-23
SSI1_MOSI
SSI1_FS
SSI1_SCK
University of Texas
Mark McDermott & Jonathan Valvano
TITLE: EE445L_Baseline_Schematic
Document Number:
REV: C
Date: September 7, 2021
Sheet: 1/1


University of Texas
Mark McDermott & Jonathan Valvano
TITLE: EE445L_Baseline_Schematic
Document Number:
REV:
Date:
Sheet: 1/1
TESTPOINT
ST7735 LCD
GND
+3.3V
RST
D/C
CARD_CS
TFT_CS
MOSI
SCK
MISO
LIGHT
ST7735_RST
ST7735_DC
ST7735_CARD_CS
ST7735_TFT_CS
ST7735_MOSI
ST7735_SCK
ST7735_MISO
C4
4.7uf
C2
.1uf
+3V3
SP1
SPEAKER
Q1
IRLD024
GND
D1
1N914ADO35-7
PB7
PWM0
PB5
TACH
I2C0_SDA
I2C0_SCK
PB1
PB0
Port B
Port A
Port F
Port C
MODE
DOWN
LEFT
RIGHT
UP
GRN_LED
BLUE_LED
RED_LED
LAUNCH_PAD_SW2

PORT Initialization

Do not use this code without confirming that it does what you need it to do.

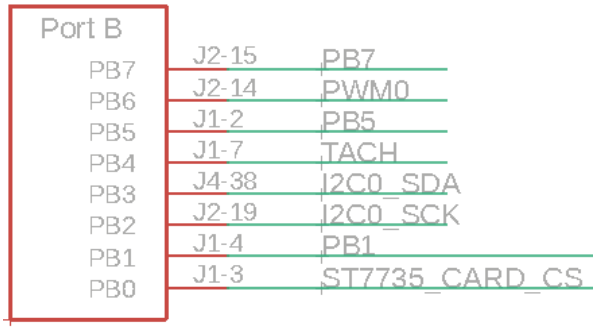
```
// ------------------------------------------------------------------------
//
// File name:     Unified_Port_Init.c
//
// Author:        Mark McDermott
// Orig gen date: July 12, 2020
// Last update:   August 17, 2020
//
// Description:   This is the unified Port initialization routine 445L Labs
//
// Usage:         Call Lab_Brd_Port_Init () if you want to initialize all ports (preferred)
//                Call the individual port inits as needed.
//
// ------------------------------------------------------------------------

#include "inc/tm4c123gh6pm.h"
#include "inc/Lab_Brd_Port_Init.h"

// --------------------------    Port_Init     ------------------------------

void Lab_Brd_Port_Init(void){
  Port_A_Init();
  Port_B_Init();
  Port_C_Init();
  Port_D_Init();
  Port_E_Init();
  Port_F_Init();
}

// ------------------------------------------------------------------------
// ----------------    PORT A Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// Port A drives the ST7735 LCD
//
// Backlight (pin 10) connected to +3.3 V
// MISO      (pin 9)  connected to PA4
// SCK       (pin 8)  connected to PA2 (SSI0Clk)
// MOSI      (pin 7)  connected to PA5 (SSI0Tx)
// TFT_CS    (pin 6)  connected to PA3 (SSI0Fss)
// CARD_CS   (pin 5)  connected to PB0 (GPIO)
// Data/CMD  (pin 4)  connected to PA6 (GPIO)
// RESET     (pin 3)  connected to PA7 (GPIO)
// VCC       (pin 2)  connected to +3.3 V
// Gnd       (pin 1)  connected to ground
//
```




```
void Port_A_Init(void){

  SYSCTL_RCGCSSI_R      |= 0x01;            // Activate SSI0

  SYSCTL_RCGCGPIO_R     |= 0x01;            // Activate port A
  while((SYSCTL_PRGPIO_R & 0x01)==0){};     // allow time for clock to start

  GPIO_PORTA_DIR_R      |= 0xC8;            // make PA3,6,7 out
  GPIO_PORTA_AFSEL_R    &= ~0xC8;           // disable alt funct on PA3,6,7
```

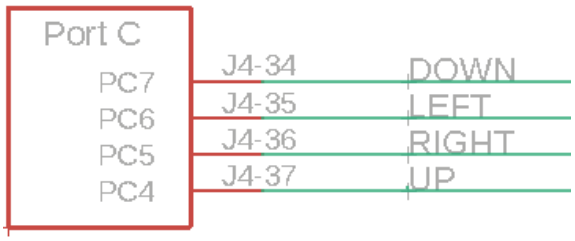
```
  GPIO_PORTA_DEN_R      |= 0xC8;           // enable digital I/O on PA3,6,7

  // configure PA3,6,7 as GPIO
  GPIO_PORTA_PCTL_R      = (GPIO_PORTA_PCTL_R
                         & 0x00FF0FFF)
                         + 0x00000000;

  GPIO_PORTA_AMSEL_R    &= ~0xC8;          // disable analog functionality on PA3,6,7

  // initialize SSI0
  GPIO_PORTA_AFSEL_R    |= 0x2C;           // enable alt funct on PA2,3,5
  GPIO_PORTA_DEN_R      |= 0x2C;           // enable digital I/O on PA2,3,5

  // configure PA2,3,5 as SSI
  GPIO_PORTA_PCTL_R      = (GPIO_PORTA_PCTL_R
                         & 0xFF0F00FF)
                         + 0x00202200;

  GPIO_PORTA_AMSEL_R    &= ~0x2C;          // disable analog functionality on PA2,3,5
  SSI0_CR1_R            &= ~SSI_CR1_SSE;   // disable SSI
  SSI0_CR1_R            &= ~SSI_CR1_MS;    // master mode

  // configure for system clock/PLL baud clock source
  SSI0_CC_R              = (SSI0_CC_R&~SSI_CC_CS_M)
                         + SSI_CC_CS_SYSPLL;

  // clock divider for 8 MHz SSIClk (80 MHz PLL/24)
  // SysClk/(CPSDVSR*(1+SCR))
  // 80/(10*(1+0)) = 8 MHz (slower than 4 MHz)
  SSI0_CPSR_R            = (SSI0_CPSR_R
                         &~SSI_CPSR_CPSDVSR_M)
                         + 10;              // must be even number

  // SCR = 0 (8 Mbps data rate)
  // SPH = 0
  // SPO = 0
  SSI0_CR0_R             &= ~(SSI_CR0_SCR_M
                         | SSI_CR0_SPH
                         | SSI_CR0_SPO);

  // FRF = Freescale format
  SSI0_CR0_R             = (SSI0_CR0_R
                         &~SSI_CR0_FRF_M)
                         + SSI_CR0_FRF_MOTO;

  // DSS = 8-bit data
  SSI0_CR0_R             = (SSI0_CR0_R
                         &~SSI_CR0_DSS_M)
                         + SSI_CR0_DSS_8;
  // enable SSI
  SSI0_CR1_R             |= SSI_CR1_SSE;

}
```

```
// ------------------------------------------------------------------------
// ----------------    PORT B Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// PB7 = GPIO/M0PWM1
// PB6 = PWM Output to Motor (M0PWM0)
// PB5 = GPIO/AIN11
// PB4 = Timer Capture input (TACH)
// PB3 = GPIO/I2C_SDA
// PB2 = GPIO/I2C0_SCL
// PB1 = GPIO
// PB0 = ST7735 Card CS
```



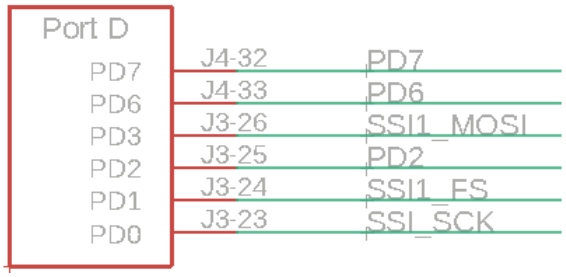

```
void Port_B_Init(void){

  SYSCTL_RCGCPWM_R      |= 0x01;            // activate PWM0

  SYSCTL_RCGCGPIO_R     |= 0x02;            // activate port B
  while((SYSCTL_PRGPIO_R & 0x02) == 0){};   // Wait

  // ---------------  Initialize PB7 as M0PWM1  ----------------------

  GPIO_PORTB_AFSEL_R    |=  0x80;           // enable alt funct on PB7
  GPIO_PORTB_PCTL_R     &= ~0xF0000000;     // configure PB7 as M0PWM1
  GPIO_PORTB_PCTL_R     |=  0x40000000;
  GPIO_PORTB_AMSEL_R    &= ~0x80;           // disable analog functionality on PB7
  GPIO_PORTB_DEN_R      |=  0x80;           // enable digital I/O on PB7

  // ---------------  Initialize PB6 as M0PWM0  ----------------------
  GPIO_PORTB_AFSEL_R    |=  0x40;           // enable alt funct on PB6
  GPIO_PORTB_PCTL_R     &= ~0x0F000000;     // configure PB6 as PWM0
  GPIO_PORTB_PCTL_R     |=  0x04000000;
  GPIO_PORTB_AMSEL_R    &= ~0x40;           // disable analog functionality on PB6
  GPIO_PORTB_DEN_R      |=  0x40;           // enable digital I/O on PB6

  // ---------------  Initialize PB5 as AIN11  -----------------------
  GPIO_PORTB_DIR_R      &= ~0x20;           // make PB5 input
  GPIO_PORTB_AFSEL_R    |=  0x20;           // enable alternate function on PB5
  GPIO_PORTB_DEN_R      &= ~0x20;           // disable digital I/O on PB5
  GPIO_PORTB_AMSEL_R    |=  0x20;           // enable analog functionality on PB5

  // ----------------- Initialize PB4 as Timer Capture input (T1CCP0) ---------
  GPIO_PORTB_DIR_R      &= ~0x10;           // make PB4 input
  GPIO_PORTB_AFSEL_R    |= 0x10;            // enable alt funct on PB4
  GPIO_PORTB_DEN_R      |= 0x10;            // enable digital I/O on PB4
                                            // configure PB4 (T1CCP0)
  GPIO_PORTB_PCTL_R     = (GPIO_PORTB_PCTL_R
                        &  0xFFF0FFFF)
                        +  0x00070000;
  GPIO_PORTB_AMSEL_R    &= ~0x10;           // disable analog functionality on PB4

  // ----------------- Initialize PB3-0 as GPIO ---------
  GPIO_PORTB_PCTL_R     &= ~0x0000FFFF;     // GPIO
  GPIO_PORTB_DIR_R      |=  0x0F;           // make PB3-0 out
  GPIO_PORTB_AFSEL_R    &= ~0x0F;           // regular port function
  GPIO_PORTB_DEN_R      |=  0x0F;           // enable digital I/O on PB3-0
  GPIO_PORTB_AMSEL_R    &= ~0x0F;           // disable analog functionality on PB3-0
  }
```

```
// ------------------------------------------------------------------------
// ----------------    PORT C Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// PC4 = UP switch
// PC5 = RIGHT switch
// PC6 = LEFT switch
// PC7 = DOWN switch
//
```



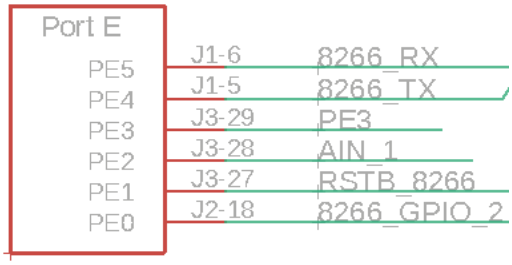

```
void Port_C_Init(void){

  SYSCTL_RCGCGPIO_R     |=  0x04;         // Activate clock for Port C
  while((SYSCTL_PRGPIO_R & 0x04) != 0x04){};  // Allow time for clock to start

  GPIO_PORTC_PCTL_R     &= ~0xFFFF0000;   // regular GPIO
  GPIO_PORTC_AMSEL_R    &= ~0xF0;         // disable analog function
  GPIO_PORTC_DIR_R      &= ~0xF0;         // inputs on PC7-PC4
  GPIO_PORTC_AFSEL_R    &= ~0xF0;         // regular port function
  GPIO_PORTC_DEN_R      |=  0xF0;         // enable digital port
  }
```

```
// ------------------------------------------------------------------------
// ----------------    PORT D Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// PD7 = GPIO/U2TX
// PD6 = GPIO/U2RX
// PD5 = Reserved for LaunchPad
// PD4 = Reserved for Launchpad
// PD3 = SSI1_MOSI (to TLV5616)
// PD2 = AIN5
// PD1 = SSI1_FS/CS (to TLV5616)
// PD0 = SSI1_SCK (to TLV5616)
```



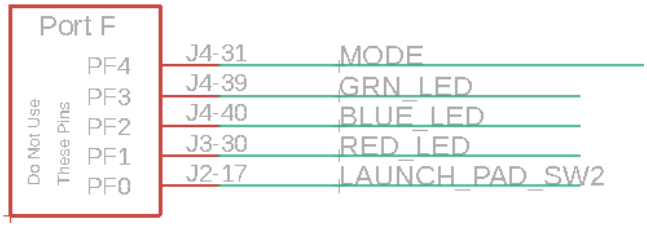

```
void Port_D_Init(void){

  SYSCTL_RCGCGPIO_R     |= 0x08;            // activate port D
  while((SYSCTL_PRGPIO_R & 0x08)==0){};     // allow time for clock to stabilize

  // ---------------  Initialize PB7 as U2TX, PB6 as U2RX  ----------------------

  GPIO_PORTD_LOCK_R      = 0x4C4F434B;      // unlock REQUIRED for PD7
  GPIO_PORTD_CR_R       |= 0xC0;            // commit PD6, PD7

  GPIO_PORTD_AMSEL_R    &= ~0xC0;           // disable analog functionality on PD6, PD7
  GPIO_PORTD_AFSEL_R    |=  0xC0;           // enable alternate function on PD6, PD7
  GPIO_PORTD_DEN_R      |=  0xC0;           // enable digital on PD6, PD7 (PD6 is U2RX, PD7 is
U2TX)
  GPIO_PORTD_PCTL_R      =(GPIO_PORTD_PCTL_R
                         & 0x00FFFFFF)
                         | 0x11000000;      // configure PD6, PD7 as UART

   // ---------------  Initialize PD2 as AIN5  ---------------------------------

  GPIO_PORTD_DIR_R      &= ~0x04;           // make PD2 input
  GPIO_PORTD_AFSEL_R    |=  0x04;           // enable alternate function on PD2
  GPIO_PORTD_DEN_R      &= ~0x04;           // disable digital I/O on PD2
  GPIO_PORTD_AMSEL_R    |=  0x04;           // enable analog functionality on PD2

  // ---------------  Initialize PD3,1,0 as SSI1 MOSI, FS & SCK  ----------------

  GPIO_PORTD_AMSEL_R    &= ~0x0B;           // disable analog functionality on PD
  GPIO_PORTD_AFSEL_R    |=  0x0B;           // enable alt funct on PD3,1,0
  GPIO_PORTD_DEN_R      |=  0x0B;           // enable digital I/O on PD3,1,0
  GPIO_PORTD_PCTL_R      = (GPIO_PORTD_PCTL_R
                          & 0xFFFF0F00)
                          + 0x00002022;
}
```

```
// ------------------------------------------------------------------------
// ----------------    PORT E Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// PE5 = 8266_RX (U5TX)
// PE4 = 8266_TX (U5RX)
// PE3 = GPIO
// PE2 = AIN_1 (Audio Input)
// PE1 = 8266_8266
// PE0 = 8266_ GPIO_2
```




```
void Port_E_Init(void){

  SYSCTL_RCGCGPIO_R     |= 0x10;            // activate port E
  while((SYSCTL_PRGPIO_R & 0x10)==0){};

 // ---------------  Initialize PE5 as U5TX, PE4 as U5RX  ----------------------

  GPIO_PORTE_AFSEL_R    |= 0x30;            // enable alt funct on PE5-4
  GPIO_PORTE_DEN_R      |= 0x30;            // enable digital I/O on PE5-4
                                            // configure PE5-4 as UART
  GPIO_PORTE_PCTL_R      = (GPIO_PORTE_PCTL_R
                         & 0xFF00FFFF)
                         + 0x00110000;
  GPIO_PORTE_AMSEL_R    &= ~0x30;           // disable analog functionality on PE

 // ---------------  Initialize PE3,1,0 as GPIO  -------------------------------

  GPIO_PORTE_DIR_R      |=  0x0A;           // output digital I/O on PE3,1
  GPIO_PORTE_DIR_R      &= ~0x01;           // input digital I/O on PE0
  GPIO_PORTE_AMSEL_R    &= ~0x0B;           // disable analog functionality on PE3,1,0
  GPIO_PORTE_AFSEL_R    &= ~0x0B;           // disable alt funct on PE3,1,0
  GPIO_PORTE_DEN_R      |=  0x0B;           // enable digital I/O on PE3,1,0

  GPIO_PORTE_PCTL_R      = (GPIO_PORTE_PCTL_R
                         & 0xFFFF0F00);

// ---------------  Initialize PE2 as AIN1  -----------------------------------

  GPIO_PORTE_DIR_R      &= ~0x04;           // make PE2 input
  GPIO_PORTE_AFSEL_R    |=  0x04;           // enable alternate function on PE2
  GPIO_PORTE_DEN_R      &= ~0x04;           // disable digital I/O on PE2
  GPIO_PORTE_AMSEL_R    |=  0x04;           // enable analog functionality on PE2
}
```

```
// ------------------------------------------------------------------------
// ----------------    PORT F Initialization   ----------------------------
// ------------------------------------------------------------------------
//
// LaunchPad Pin Assignments
// PF4 = SW1
// PF3 = GREEN_LED
// PF2 = BLUE_LED
// PF1 = RED_LED
// PF0 = SW2
```




```
void Port_F_Init(void){

  SYSCTL_RCGCGPIO_R     |= 0x00000020;      // activate clock for Port F
  while((SYSCTL_PRGPIO_R & 0x20)==0){};     // allow time for clock to stabilize

  GPIO_PORTF_LOCK_R     = 0x4C4F434B;       // unlock GPIO Port F
  GPIO_PORTF_CR_R       = 0x1F;             // allow changes to PF4-0

  GPIO_PORTF_AMSEL_R    = 0x00;             // disable analog on PF
  GPIO_PORTF_PCTL_R     = 0x00000000;       // PCTL GPIO on PF4-0
  GPIO_PORTF_DIR_R      = 0x0E;             // PF4,PF0 in, PF3-1 out
  GPIO_PORTF_AFSEL_R    = 0x00;             // disable alt function on PF7-0
  GPIO_PORTF_PUR_R      = 0x11;             // enable pull-up on PF0 and PF4
  GPIO_PORTF_DEN_R      = 0x1F;             // enable digital I/O on PF4-0
}
```