**GPIO Exercise Solutions**

**Issue 1.0**

# Exercises (With Embedded Spreadsheets)

1. What are the valid input voltage ranges for a KL25 MCU with VDD = 3 V? With 2V?



1. Examine the schematic for your Freedom KL25Z board. How many GPIO port bits are available for Port A? Port B? Port C? Port D? Port E?

Port A: 8 or 9 (if RESET/PTA20 is included)

Port B: 10

Port C: 16

Port D: 8 or 9 (if SDA\_PTD5 is included)

Port E: 13

1. Which digital outputs on the KL25 subfamily support high drive capability? Refer to the KL25 Sub-Family Data Sheet.

Port B 0, Port B 1, Port D 6, Port D7

1. Calculate the resistor values needed to limit current through the blue and red LEDs of Figure 8 to 18 mA each. Assume the supply voltage is 3 V.



1. Use a multimeter to measure the actual VDD on your MCU board.

VDD = Approximately 3.3 V

1. Modify the LCD driver code presented in this chapter to use PORTA bits 10-13 for the data bus (DB4-7) and bits 20-22 for control lines E, R/~W and RS respectively.

// lcd\_4bit.h

#define PIN\_DATA\_SHIFT ( **10** )

#define PIN\_E\_SHIFT ( **20** )

#define PIN\_E ( 1 << PIN\_E\_SHIFT)

#define PIN\_RW\_SHIFT ( PIN\_E\_SHIFT + 1 )

#define PIN\_RW ( 1 << PIN\_RW\_SHIFT)

#define PIN\_RS\_SHIFT ( PIN\_E\_SHIFT + 2)

#define PIN\_RS ( 1 << PIN\_RS\_SHIFT)

#define PINS\_CTRL (0x07 << PIN\_E\_SHIFT)

#define PINS\_DATA (0x0F << PIN\_DATA\_SHIFT)

#define PINS\_ALL (PINS\_CTRL | PINS\_DATA)

/\* Enable Clock for peripheral driving LCD pins \*/

#define LCD\_CLOCK\_EN SIM->SCGC5 |= **SIM\_SCGC5\_PORTA\_MASK**;

/\* pin E setting to 0 or 1 \*/

#define LCD\_E(x) PTA->PDOR = (PTA->PDOR & ~PIN\_E) | ((x) ? PIN\_E : 0);

/\* pin RW setting to 0 or 1 \*/

#define LCD\_RW(x) PTA->PDOR = (PTA->PDOR & ~PIN\_RW) | ((x) ? PIN\_RW : 0);

/\* pin RS setting to 0 or 1 \*/

#define LCD\_RS(x) PTA->PDOR = (PTA->PDOR & ~PIN\_RS) | ((x) ? PIN\_RS : 0);

/\* Reading DATA pins \*/

#define LCD\_DATA\_IN (((PTA->PDIR & PINS\_DATA) >> PIN\_DATA\_SHIFT) & 0x0F)

/\* Writing value to DATA pins \*/

#define LCD\_DATA\_OUT(x) PTA->PDOR = (PTA->PDOR & ~PINS\_DATA) | ((x) << PIN\_DATA\_SHIFT);

/\* Setting all pins to output mode \*/

#define LCD\_ALL\_DIR\_OUT PTA->PDDR = PTA->PDDR | PINS\_ALL;

/\* Setting DATA pins to input mode \*/

#define LCD\_DATA\_DIR\_IN PTA->PDDR = PTA->PDDR & ~PINS\_DATA;

/\* Setting DATA pins to output mode \*/

#define LCD\_DATA\_DIR\_OUT PTA->PDDR = PTA->PDDR | PINS\_DATA;

----------------------------------------------------------------------------------------

// lcd\_4bit.c

void lcd\_init (void)

{

LCD\_CLOCK\_EN; /\* Enable clock for peripheral \*/

/\* Set Pin Mux to GPIO \*/

PORTA->PCR[PIN\_DATA\_SHIFT] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_DATA\_SHIFT+1] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_DATA\_SHIFT+2] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_DATA\_SHIFT+3] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_E\_SHIFT] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_RW\_SHIFT] = PORT\_PCR\_MUX(1);

PORTA->PCR[PIN\_RS\_SHIFT] = PORT\_PCR\_MUX(1);