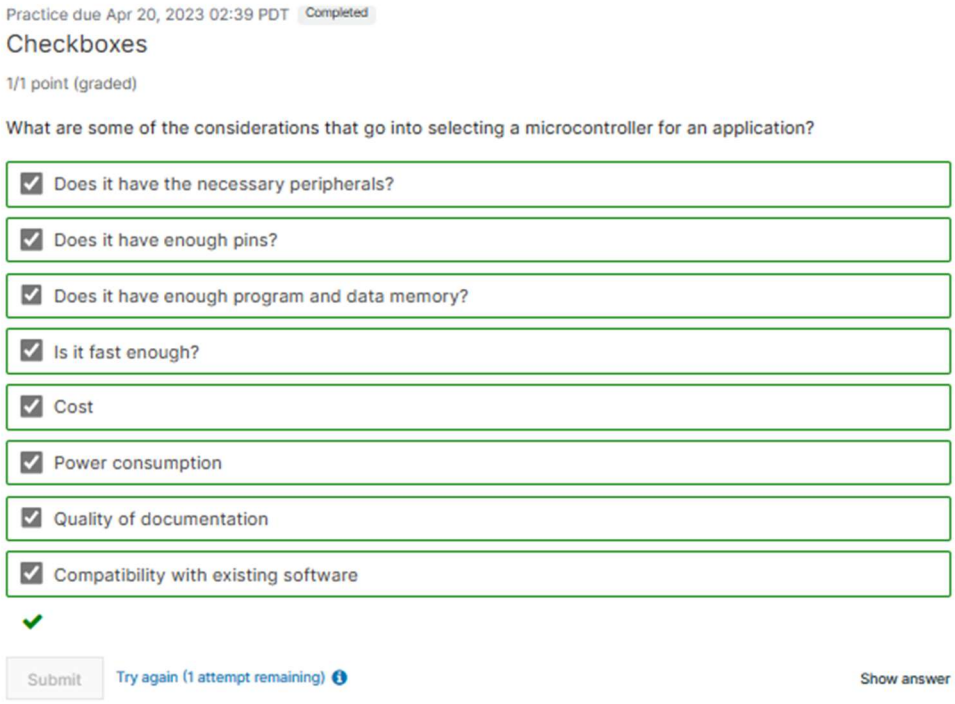
❖ Chapter 9: Embedded Systems / Introduction



Practice due Apr 20, 2023 02:39 PDT Completed

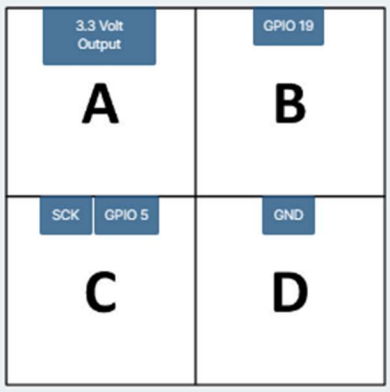
Consider the SparkFun RED-V Thing Plus board with pins A-D annotated below.

Thing Plus Board Pinout

1/1 point (graded)

Keyboard Help

Drag the pin functions to the letter labeling a pin in the photo above.



Reset

FEEDBACK

Good work! You can identify pins on the Thing Plus board.

Thing Plus Board

1/1 point (graded)

On the RED-V Thing Plus Board, GPIO 5 can be used to (check all that apply)

- [x] drive a pin on the board high or low
- [x] measure whether the pin is high or low
- [x] turn ON or OFF a blue LED
- [x] be a serial clock for a SPI interface
- [ ] electrocute yourself

Submit Try again (1 attempt remaining) Show answer

Now that you've learned to access GPIO registers, refer to the following excerpts from the FE310 manual to figure out how to access UART registers.

FE310 Memory Map

Base	Top	Attr.	Description	Notes
0x0000_0000	0x0000_0FFF	RWX A	Debug	Debug Address Space
0x0000_1000	0x0000_1FFF	R XC	Mode Select	On-Chip Non Volatile Memory
0x0000_2000	0x0000_2FFF		Reserved	
0x0000_3000	0x0000_3FFF	RWX A	Error Device	
0x0000_4000	0x0000_FFFF		Reserved	
0x0001_0000	0x0001_1FFF	R XC	Mask ROM (8 KiB)	
0x0001_2000	0x0001_FFFF		Reserved	
0x0002_0000	0x0002_1FFF	R XC	OTP Memory Region	
0x0002_2000	0x001F_FFFF		Reserved	
0x0200_0000	0x0200_FFFF	RW A	CLINT	On-Chip Peripherals
0x0201_0000	0x07FF_FFFF		Reserved	
0x0800_0000	0x0800_1FFF	RWX A	E31 ITIM (8 KiB)	
0x0800_2000	0x0BFF_FFFF		Reserved	
0x0C00_0000	0x0FFF_FFFF	RW A	PLIC	
0x1000_0000	0x1000_0FFF	RW A	AON	
0x1000_1000	0x1000_7FFF		Reserved	
0x1000_8000	0x1000_8FFF	RW A	PRCI	
0x1000_9000	0x1000_FFFF		Reserved	
0x1001_0000	0x1001_0FFF	RW A	OTP Control	
0x1001_1000	0x1001_1FFF		Reserved	
0x1001_2000	0x1001_2FFF	RW A	GPIO	
0x1001_3000	0x1001_3FFF	RW A	UART 0	
0x1001_4000	0x1001_4FFF	RW A	QSPI 0	
0x1001_5000	0x1001_5FFF	RW A	PWM 0	
0x1001_6000	0x1001_6FFF	RW A	I2C 0	
0x1001_7000	0x1002_2FFF		Reserved	
0x1002_3000	0x1002_3FFF	RW A	UART 1	
0x1002_4000	0x1002_4FFF	RW A	SPI 1	
0x1002_5000	0x1002_5FFF	RW A	PWM 1	
0x1002_6000	0x1003_3FFF		Reserved	
0x1003_4000	0x1003_4FFF	RW A	SPI 2	
0x1003_5000	0x1003_5FFF	RW A	PWM 2	
0x1003_6000	0x1FFF_FFFF		Reserved	
0x2000_0000	0x3FFF_FFFF	R XC	QSPI 0 Flash (512 MiB)	Off-Chip Non-Volatile Memory
0x4000_0000	0x7FFF_FFFF		Reserved	
0x8000_0000	0x8000_3FFF	RWX A	E31 DTIM (16 KiB)	On-Chip Volatile Memory
0x8000_4000	0xFFFF_FFFF		Reserved	

Offset	Name	Description
0x00	txdata	Transmit data register
0x04	rxdata	Receive data register
0x08	txctrl	Transmit control register
0x0C	rxctrl	Receive control register
0x10	ie	UART interrupt enable
0x14	ip	UART interrupt pending
0x18	div	Baud rate divisor

Table 55: Register offsets within UART memory map

Transmit Data Register (txdata)				
Register Offset		0x0		
Bits	Field Name	Attr.	Rst.	Description
[7:0]	data	RW	X	Transmit data
[30:8]	Reserved			
31	full	RO	X	Transmit FIFO full

Table 56: Transmit Data Register

Transmit Control Register (txctrl)				
Register Offset		0x8		
Bits	Field Name	Attr.	Rst.	Description
0	txen	RW	0x0	Transmit enable
1	nstop	RW	0x0	Number of stop bits
[15:2]	Reserved			
[18:16]	txcnt	RW	0x0	Transmit watermark level
[31:19]	Reserved			

Table 58: Transmit Control Register

UART1 Base Address

1/1 point (graded)

What is the base address for the UART1? Express your answer as a hexadecimal number with a leading 0x and an underscore in the middle, like 0×1003_6000.

0×1002_3000 ✔

Submit Try again (1 attempt remaining) Show answer

Pointer

1/1 point (graded)

Which line of code correctly declares a pointer to the 32-bit UART1 txctrl register?

- ○ volatile uint32_t *UART1_TXCTRL = (uint32_t*)0×10013000;
- ○ volatile uint32_t *UART1_TXCTRL = (uint32_t*)0×10023000;
- ◉ volatile uint32_t *UART1_TXCTRL = (uint32_t*)0×10023008;
- ○ volatile bit *UART1_TXCTRL_TXEN = (bit*)0×10023000;

✔

Submit Try again (1 attempt remaining) Show answer

In the following questions, let the following expressions substitute for normal C syntax, which EdX can't reproduce in these questions:

- shl = <<
- shr = >>
- and = &

Bit Setting

1/1 point (graded)

Which line of code correctly enables the UART1 transmitter without disturbing any of the other control bits?

- ○ *UART1_TXCTRL = 1;
- ○ *UART1_TXCTRL0 = 1;
- ○ *UART1_TXCTRL = 1 shl 0;
- ◉ *UART1_TXCTRL0 |= (1 shl 0);
- ○ *UART1_TXCTRL0 |= (0 shl 1);

✔

Submit Try again (1 attempt remaining) Show answer

Waiting

1/1 point (graded)

Suppose UART1_TXDATA is declared as a pointer to 0×10013000. What line of code will wait until the transmit data register is not full?

- ◉ while ((*UART1_TXDATA shr 31) and 1);
- ○ while ((*UART1_TXDATA shr 32) and 1);
- ○ while (!(*UART1_TXDATA shr 32) and 1);
- ○ while (*UART1_TXDATA_FULL);
- ○ while (*UART1_TXDATA[31]);

✔

Submit Try again (1 attempt remaining) Show answer

Practice due Apr 20, 2023 02:39 PDT Completed

Suppose the highlighted line were commented out in the pinMode function. What would be different, if anything?

```
void pinMode(int gpio_pin, int function) {
  switch(function) {
    case INPUT:
      GPIO0->input_en   |= (1 << gpio_pin);  // Sets a pin as an input
//    GPIO0->output_en  &= ~(1 << gpio_pin); // Disable output
      GPIO0->iof_en     &= ~(1 << gpio_pin); // Disable IOF
      break;
    case OUTPUT:
      GPIO0->output_en  |= (1 << gpio_pin);  // Set pin as an output
      GPIO0->input_en   &= ~(1 << gpio_pin); // Disable input
      GPIO0->iof_en     &= ~(1 << gpio_pin); // Disable IOF
      break;
  }
}
```

pinMode

1/1 point (graded)

Check all that apply:

- [ ] The line is unnecessary and the function is better without it.
- [x] The first time pinMode is called for a pin, it will work normally.
- [x] If a program first set a pinMode to OUTPUT, and later changed the pinMode to INPUT, the pin will continue to behave as an output.
- [ ] The pin will take on its special function instead.

✔

Submit Try again (1 attempt remaining) Show answer

Practice due Apr 20, 2023 02:39 PDT Completed

Timer Overflow

1/1 point (graded)

How long will it take for the mtime register to overflow?

- ○ 2^17 us
- ○ 2^49 ms
- ◉ 2^49 seconds

✔

Submit Try again (1 attempt remaining) Show answer

Practice due Apr 20, 2023 02:39 PDT Completed

Parameterized Delay

1/1 point (graded)

What are benefits of expressing delay in terms of DUR rather than as absolute numbers in the code. Check all that apply.

- [ ] The delay function can only take a #define because its input is an int, not a uint64_t, and it lacks a prototype declared in the header.
- [x] It is easy to change the tempo of the output by changing only one number.
- [x] It emphasizes the relative delays of the pulses.
- [ ] To dereference the structure point in logarithmic time using far jumps and abstract data types while facilitating real-time garbage collection and type isomorphism as our resources were tasked to do.
- [ ] Quidquid latine dictum sit, altum sonatur.

✔

Submit Try again (1 attempt remaining) Show answer

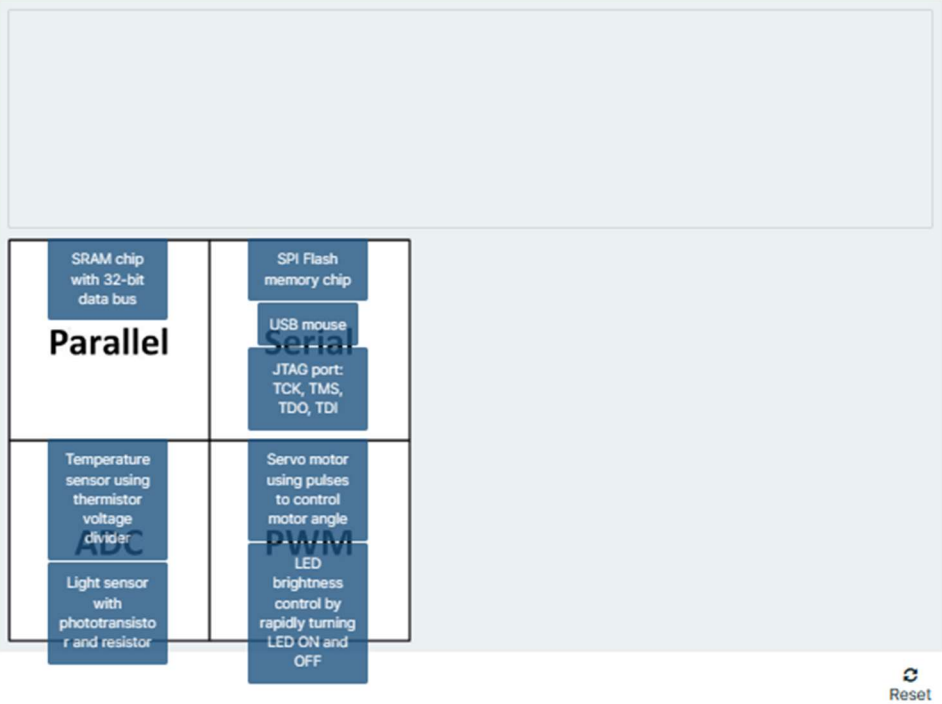
Practice due Apr 20, 2023 02:39 PDT Completed

interfacing

1/1 point (graded)

Keyboard Help

Drag and drop to indicate the best way to interface various devices to a micro controller.



Practice due Apr 20, 2023 02:39 PDT Completed

Baud Rate

1/1 point (graded)

If the sckdiv register contains 79, what frequency will SCK operate at? Express your answer in Hz.

100000 ✔

100000

Submit Try again (1 attempt remaining) Show answer

Register Address

1/1 point (graded)

What is the address of the SPI1 RXDATA register? Express your answer as an 8-digit hexadecimal number with no leading 0x.

0×1002404C ✔

Submit Try again (1 attempt remaining) Show answer

Suppose we declare:

volatile uint32_t *SPI1_TXDATA = (uint32_t*)0×10024000;

int full;

Consider the following code:

(a) full = *SPI1_TXDATA >> 31;

(b) full = (*SPI_TXDATA >> 31) & 0×1;

(c) full = *SPI1_TXDATA |= (1 << 31);

(d) while ((*SPI1_TXDATA >> 31) & 0×1) full = 1;

SPI1 TXDATA FULL

1/1 point (graded)

Which of the lines of code above correctly return the SPI1 TXDATA FULL bit without relying on bitfields? Choose all that apply.

- [x] a
- [x] b
- [ ] c
- [ ] d

✔

Submit Try again (1 attempt remaining) Show answer

Practice due Apr 20, 2023 02:39 PDT Completed

Connections

1/1 point (graded)

How many wires does the LIS3DH need to have connected to the RED-V, including signals and power?

6 ✔

6

Submit Try again (1 attempt remaining) Show answer

Accelerometer Register Addresses

1/1 point (graded)

What is the relationship between register addresses on the accelerometer and on the microcontroller?

- (●) There is no relationship. All accelerometer registers are accessed by reading or writing the same SPI1_TXDATA and SPI_RXDATA registers. The address is specified in the first 8 bits transmited over TXDATA.
- ( ) The desired address of the accelerometer register is added to the base address of the SPI memory-mapped I/O peripheral.
- ( ) The desired address of the accelerometer register is shifted left by 2 and then added to the SPI1_TXDATA register address.
- ( ) The desired address of the accelerometer register is multiplied by pi, added to the address of the SPI1_BS register, and divided by the speed of light.

✔

Submit Try again (1 attempt remaining) Show answer

WHO_AM_I Register

1/1 point (graded)

If you read the accelerometer's WHO_AM_I register at address 0F, what should you expect to get back if the accelerometer is correctly wired to the microcontroller? Express your answer in decimal.

51 ✔

51

Submit Try again (1 attempt remaining) Show answer

Practice due Apr 30, 2023 23:59 PDT Completed

Design Principles

1/1 point (graded)

What are the four design principles guiding the RISC-V architecture?

- [x] Simplicity favors regularity
- [ ] Make the assembly language resemble the high level language
- [x] Make the common case fast
- [x] Smaller is faster
- [ ] More instructions make for a faster computer
- [x] Good design demands good compromise
- [ ] Semper ubi sub ubi

✔

Submit Try again (1 attempt remaining) Show answer