

1 Consider the AVR code segment shown below that initializes I/O and interrupts for Tekbot shown below.

```

1 ldi mpr, (1<<EngEnL)|(1<<EngEnR)|(1<<EngDirR)|(1<<EngDirL)
2 out DDRA, mpr
3 ldi mpr, (0<<WskrL)|(0<<WskrR)
4 out DDRD, mpr
5 ldi mpr, $FF
6 out PORTD, mpr
7 ldi mpr, (1<<ISC01)|(1<<ISC00)|(1<<ISC11)|(1<<ISC10)
8 sts EICRA, mpr
9 ldi mpr, (1<<INT0)|(1<<INT1)
10 out EIMSK, mpr

```

Lines 1 and 2 set the data direction register to control the engine enable and engine direction

Lines 3 and 4 set the data direction register x to detect for the left and right whiskers

Lines 5 and 6 enable pull up resistors

Lines 7 and 8 set interrupt sense control to detect whisker movements on falling edges

Lines 9 and 10 enable interrupts

2 Rewrite the WAIT subroutine so that it waits for 1 sec using the 16bit timer/counter 1. Assume the system clock is 16Mhz and timer/counter1 is operating under normal mode. This is done by doing the following. A) Timer/counter 1 is initialized to operate in normal mode. B) The wait subroutine loads the proper value into TCNT1 waits until TOV1 is set. Once TOV1 is set it is cleared and the wait returns

```
.include "m128def.inc"
```

```
.def mpr = r16
```

```
...
```

```
.ORG $ 0000
```

RJMP Initialize

.ORG \$0046 ;End of interrupt vectors

Initialize:

SBI DDRB, PB4 ;set OC0 (pin 4) on port b for output

LDI A, 0b00000111 ; setting timer normal mode with prescaler

OUT TCCR1, A ; load data into counter control register

WAIT:

LDI r17, 100 ;C Load Count = 100

WAIT\_10msec

LDI r16, 99 ; Value for delay

OUT TCNT1, r16

LOOP:

In r18, TIFR

ANDI r18, 0b00000001

BREQ LOOP

LDI r18, 0b00000001

OUT TIFR, r18

DEC r17

BRNE WAIT\_10msec

Ret

3

1. Ldi mpr (1<<COM00) |(1<<COM00) |(1<<COM00) | (1<<COM00)
2. Out TCCRO mpr
- 3.

4Write initUSATY1 to configure atmega128 usart0 to operate as a transmitter and send a byte of data every time usart0 data register empty interrupt occurs. The transmitter operates with the following settings

8 bits, 1 stop bits, and odd parity

9600 baud rate, double data rate

Transmitter enabled

Normal asynchronous mode operation

Interrupt enabled

```
.include "m128def.inc"
```

```
.def mpr = r16
```

```
.ORG $0000
```

```
    RJMP Init
```

```
    ...
```

```
.ORG $003E
```

```
    RCALL SendData
```

```
    RETI
```

```
    ...
```

```
.ORG $0046
```

```
Init:
```

```
    ...Initialize stack...
```

```
    RCALL initUSART
```

```
Main:
```

```
    RCALL SendData
```

```
Loop:
```

RJMP Loop

initUSART0:

;init I/O ports ; The next two lines init port d for output.

Ldi mpr, 0b00001000

Out DDRD, mpr

;init usart0 ; the next two times init the usart for transmitting and double the data rate

Ldi mpr, (1<<U2X1)

sts USCR0A, mpr

;set baudrate at 9600 ;set the baudrate

Ldi mpr, 103

sts UBRR0L, mpr

;set frame format for 8 data bits 1 stop bits and odd parity

Ldi mpr, (1<<USBS1 | 1<<UPM11 | 1<<UCSZ11 | 0<<UCSZ10)

sts UCSR0C, mpr

;enable data register empty interrupts

Ldi mpr, (1<<UDRIE0)

sts UCSR0B, mpr

ret

SendData:

ld r17, X+ ; Assume X points to the data to be transmitted

sts UDR0, r17 ; Move data to Transmit Data Buffer

ret