Willis T. Allstead
Professor Egbert
CPE 301
19 October 2016

HW 6

1)

2) Since both question 1 & 2 are basically one question I will answer both here.

For the Initialization function:
(All declarations in final compilation at the end of this report)

```
void U0init(long U0baud) {
  unsigned long FCPU = 16000000;
  unsigned int tbaud;
  tbaud = (FCPU / 16 / U0baud - 1);
  *myUCSR0A = 0x20;
  *myUCSR0B = 0x18;
  *myUCSR0C = 0x06;
  *myUBRR0  = tbaud;
}
```

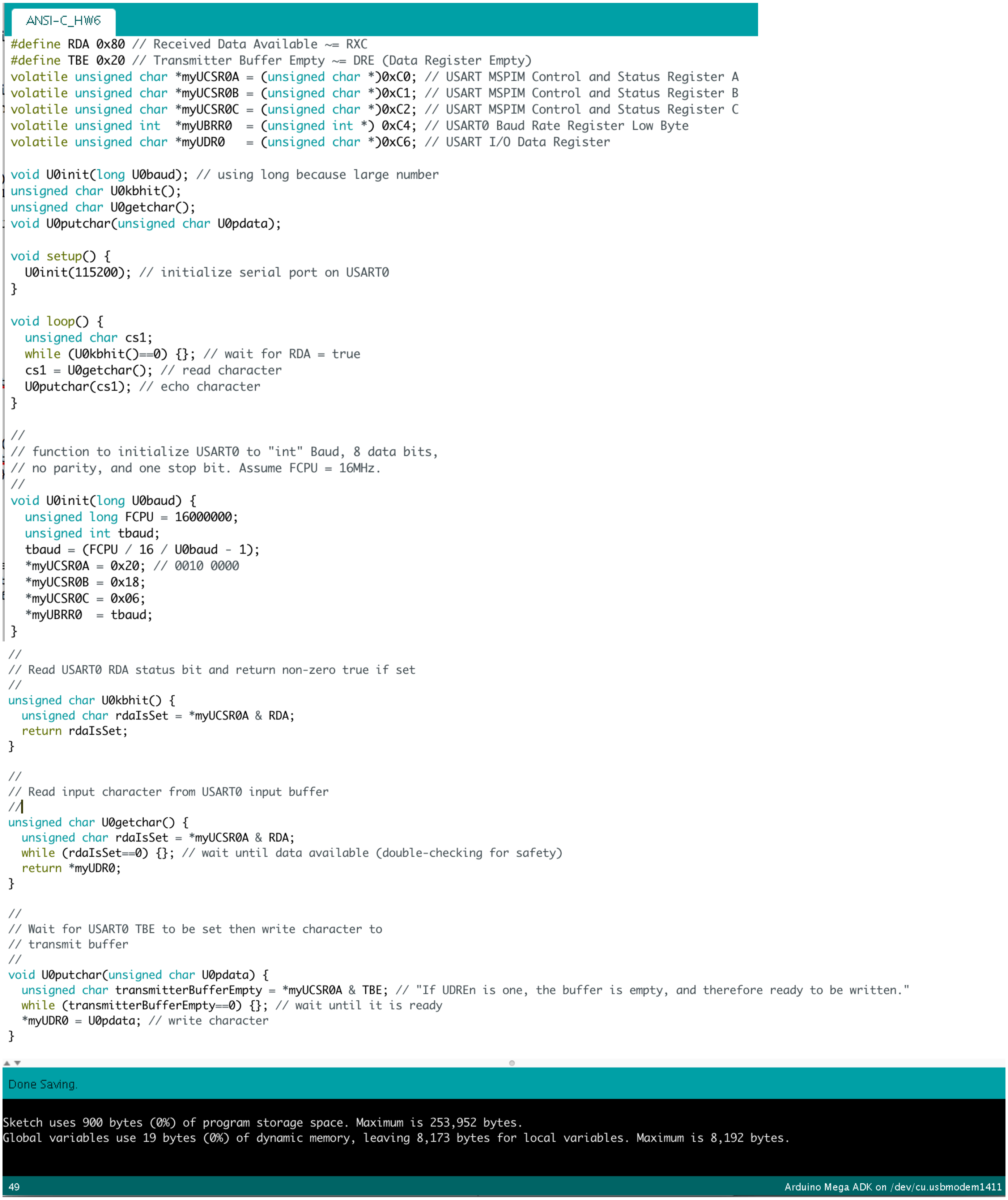
3)

```
unsigned char U0kbhit() {
  unsigned char rdaIsSet = *myUCSR0A & RDA;
  return rdaIsSet;
}

unsigned char U0getchar() {
  unsigned char rdaIsSet = *myUCSR0A & RDA;
  while (rdaIsSet==0) {}; // wait until data available (double-checking for
safety)
  return *myUDR0;
}

void U0putchar(unsigned char U0pdata) {
// "If UDREn is one, the buffer is empty, and therefore ready to be written."
  unsigned char transmitterBufferEmpty = *myUCSR0A & TBE;
  while (transmitterBufferEmpty==0) {}; // wait until it is ready
  *myUDR0 = U0pdata; // write character
}
```

Final Code Compilation)



```
#define RDA 0x80 // Received Data Available ~= RXC
#define TBE 0x20 // Transmitter Buffer Empty ~= DRE (Data Register Empty)
volatile unsigned char *myUCSR0A = (unsigned char *)0xC0; // USART MSPIM Control and Status Register A
volatile unsigned char *myUCSR0B = (unsigned char *)0xC1; // USART MSPIM Control and Status Register B
volatile unsigned char *myUCSR0C = (unsigned char *)0xC2; // USART MSPIM Control and Status Register C
volatile unsigned int  *myUBRR0  = (unsigned int *) 0xC4; // USART0 Baud Rate Register Low Byte
volatile unsigned char *myUDR0   = (unsigned char *)0xC6; // USART I/O Data Register

void U0init(long U0baud); // using long because large number
unsigned char U0kbhit();
unsigned char U0getchar();
void U0putchar(unsigned char U0pdata);

void setup() {
  U0init(115200); // initialize serial port on USART0
}

void loop() {
  unsigned char cs1;
  while (U0kbhit()==0) {}; // wait for RDA = true
  cs1 = U0getchar(); // read character
  U0putchar(cs1); // echo character
}

//
// function to initialize USART0 to "int" Baud, 8 data bits,
// no parity, and one stop bit. Assume FCPU = 16MHz.
//
void U0init(long U0baud) {
  unsigned long FCPU = 16000000;
  unsigned int tbaud;
  tbaud = (FCPU / 16 / U0baud - 1);
  *myUCSR0A = 0x20; // 0010 0000
  *myUCSR0B = 0x18;
  *myUCSR0C = 0x06;
  *myUBRR0  = tbaud;
}

//
// Read USART0 RDA status bit and return non-zero true if set
//
unsigned char U0kbhit() {
  unsigned char rdaIsSet = *myUCSR0A & RDA;
  return rdaIsSet;
}

//
// Read input character from USART0 input buffer
//
unsigned char U0getchar() {
  unsigned char rdaIsSet = *myUCSR0A & RDA;
  while (rdaIsSet==0) {}; // wait until data available (double-checking for safety)
  return *myUDR0;
}

//
// Wait for USART0 TBE to be set then write character to
// transmit buffer
//
void U0putchar(unsigned char U0pdata) {
  unsigned char transmitterBufferEmpty = *myUCSR0A & TBE; // "If UDREn is one, the buffer is empty, and therefore ready to be written."
  while (transmitterBufferEmpty==0) {}; // wait until it is ready
  *myUDR0 = U0pdata; // write character
}
```

Done Saving.

```
Sketch uses 900 bytes (0%) of program storage space. Maximum is 253,952 bytes.
Global variables use 19 bytes (0%) of dynamic memory, leaving 8,173 bytes for local variables. Maximum is 8,192 bytes.
```

49 Arduino Mega ADK on /dev/cu.usbmodem1411