Monday 7 March 2022

ELEC-C7420 Basic Principles in Networking Spring 2022

Assignment III: Introduction to basic cryptography using Arduino

PSALTAKIS GEORGIOS

Aalto University
School of Engineering

Goals of the experiment

The goals of the experiment are the introduction to the basics of cryptography. By doing this experiment we are able to understand how does encryption works on a basic level and as well how to decrypt a basic ciphertext.

For this experiment, we are implementing RSA encryption in an Arduino board and we are implementing encryption and decryption of the text by using the console serial and options 1 or 2 before the encrypted text or plain text regarding the option.

Experimental Setup

The setup of the experiment is simple. We have implemented the code and we give a simple help text for the user
Enter 1 to encrypt or 2 to decrypt.

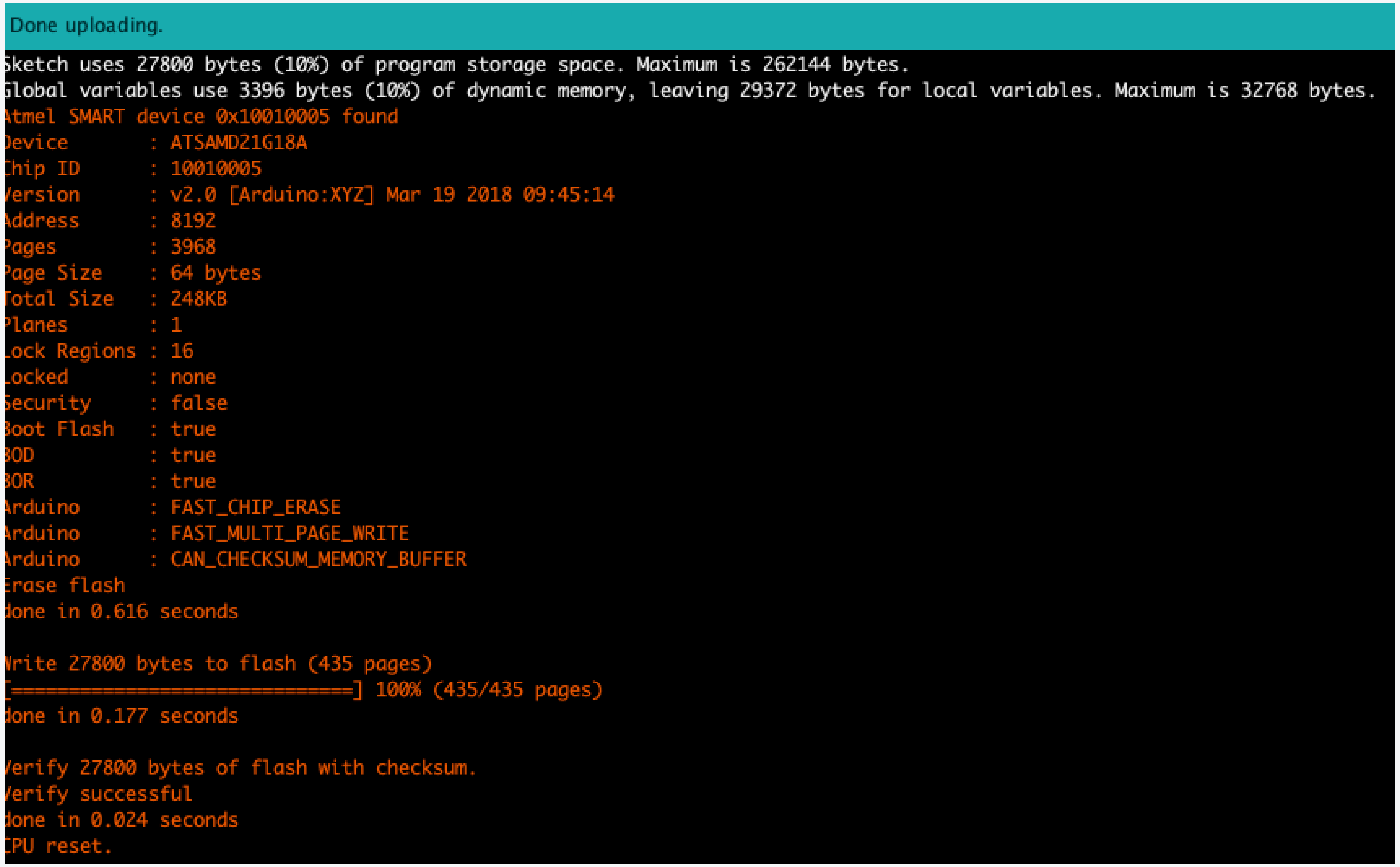
This code has been uploaded to the Arduino and using the serial input we can request if we want to encrypt or decrypt the code we want.

Then we input the text "1 hello world" to get the encrypted version of the code and then we get the decrypt the encrypted version of hello world by using " 2   17.4.11.11.24.-30.27.24.22.11.8.-23."

This outputs Decrypted: hello world

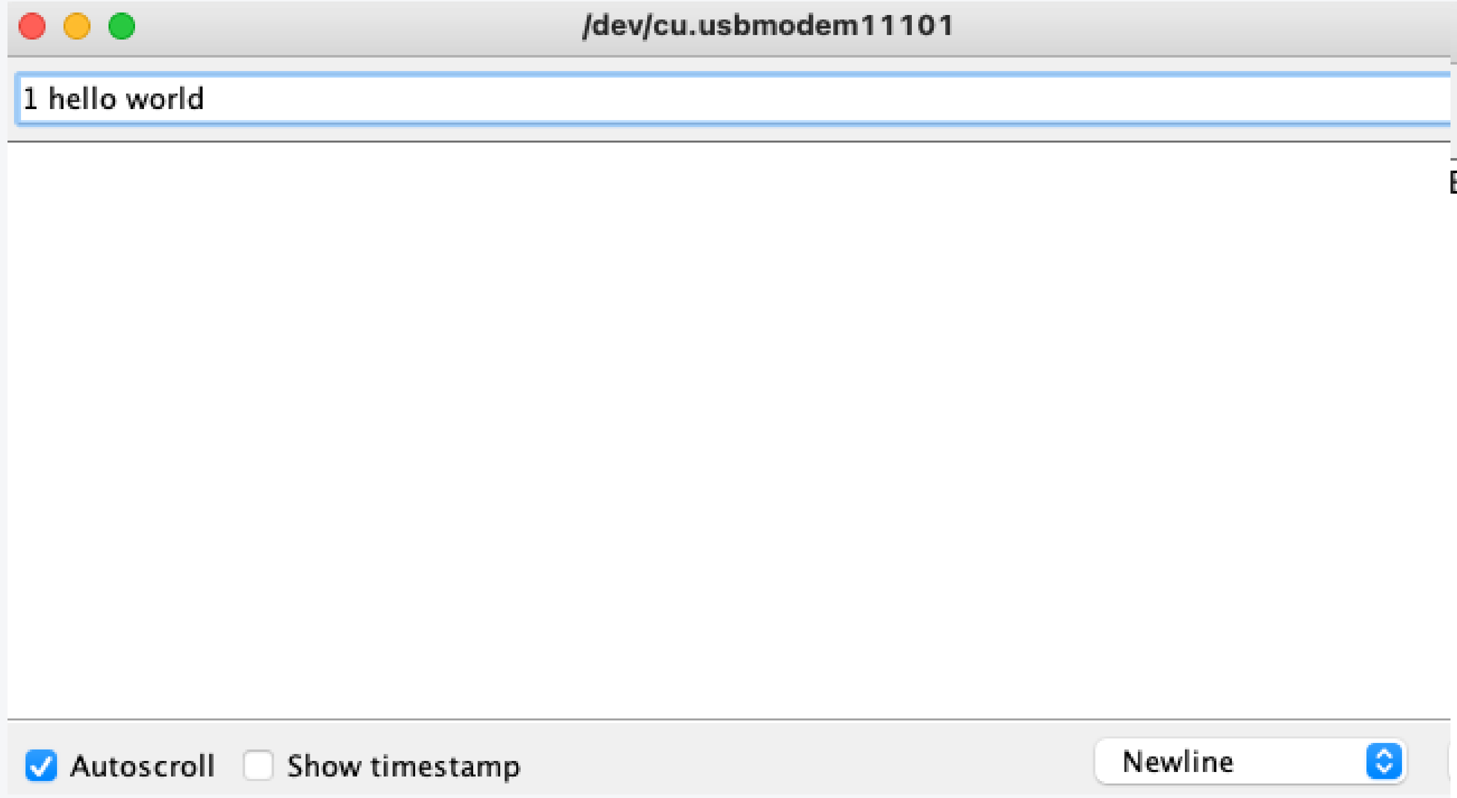
Results & Conclusion

Creating successful compilation of sketch and upload without an error



Done uploading.

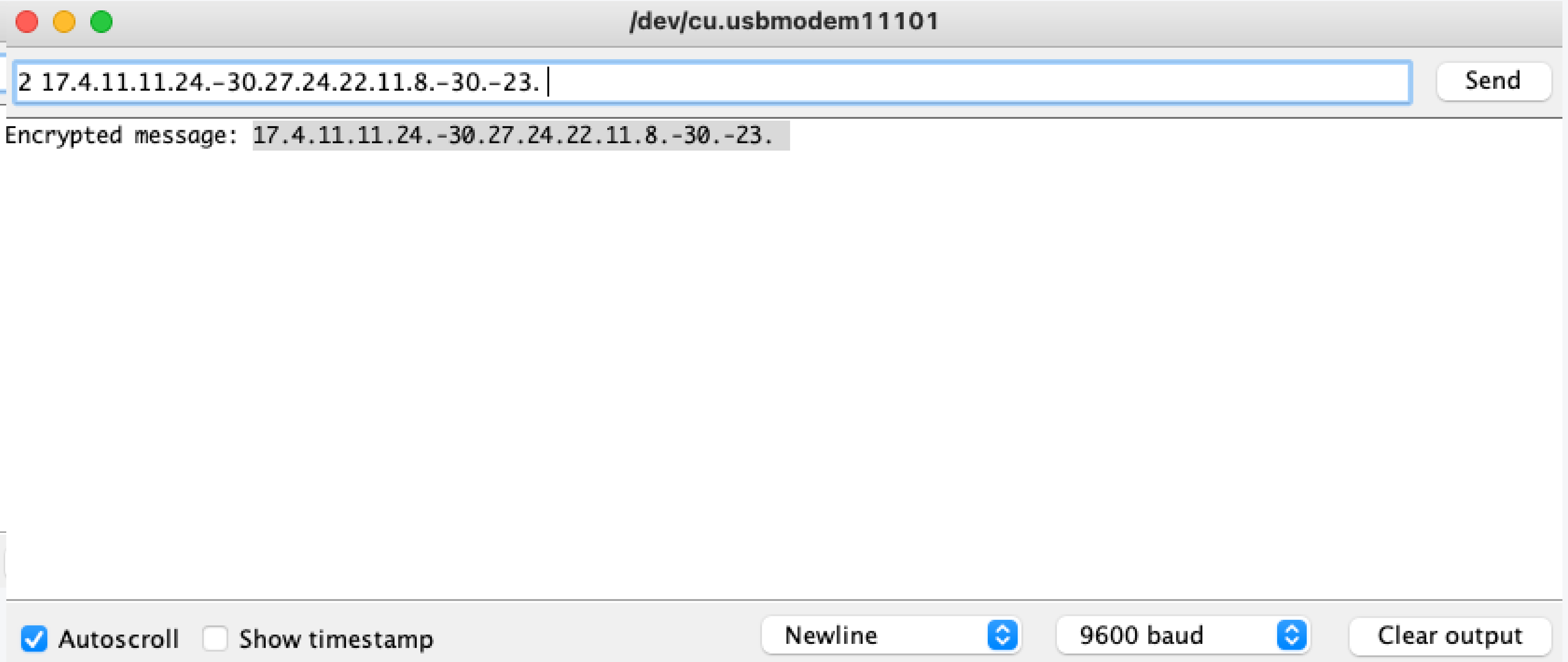
```
Sketch uses 27800 bytes (10%) of program storage space. Maximum is 262144 bytes.
Global variables use 3396 bytes (10%) of dynamic memory, leaving 29372 bytes for local variables. Maximum is 32768 bytes.
Atmel SMART device 0x10010005 found
Device       : ATSAMD21G18A
Chip ID      : 10010005
Version      : v2.0 [Arduino:XYZ] Mar 19 2018 09:45:14
Address      : 8192
Pages        : 3968
Page Size    : 64 bytes
Total Size   : 248KB
Planes       : 1
Lock Regions : 16
Locked       : none
Security     : false
Boot Flash   : true
BOD          : true
BOR          : true
Arduino      : FAST_CHIP_ERASE
Arduino      : FAST_MULTI_PAGE_WRITE
Arduino      : CAN_CHECKSUM_MEMORY_BUFFER
Erase flash
done in 0.616 seconds

Write 27800 bytes to flash (435 pages)
[==============================] 100% (435/435 pages)
done in 0.177 seconds

Verify 27800 bytes of flash with checksum.
Verify successful
done in 0.024 seconds
CPU reset.
```



/dev/cu.usbmodem11101

```
1 hello world
```

☑ Autoscroll ☐ Show timestamp Newline

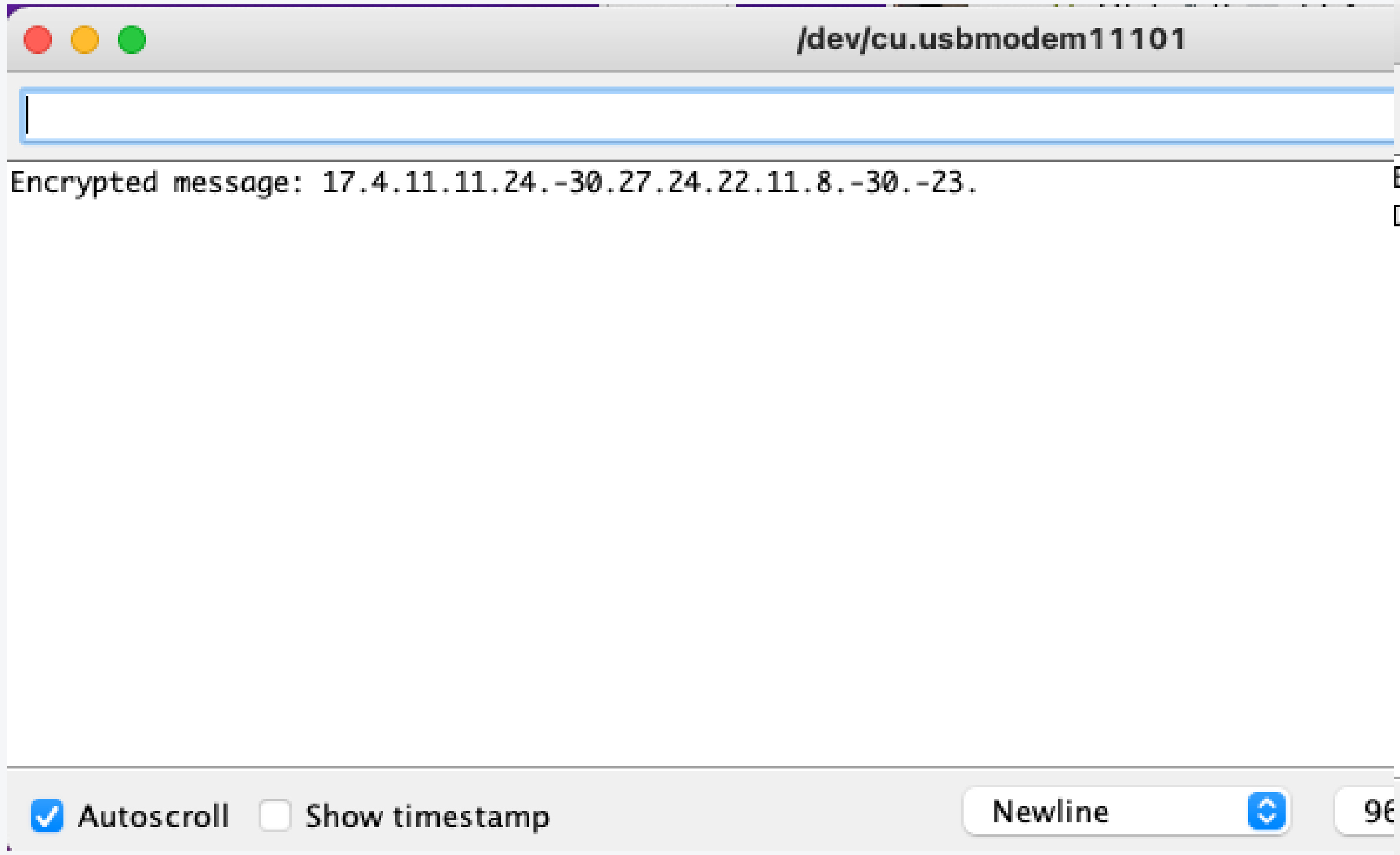


/dev/cu.usbmodem11101

```
2 17.4.11.11.24.-30.27.24.22.11.8.-30.-23.
```

```
Encrypted message: 17.4.11.11.24.-30.27.24.22.11.8.-30.-23.
```

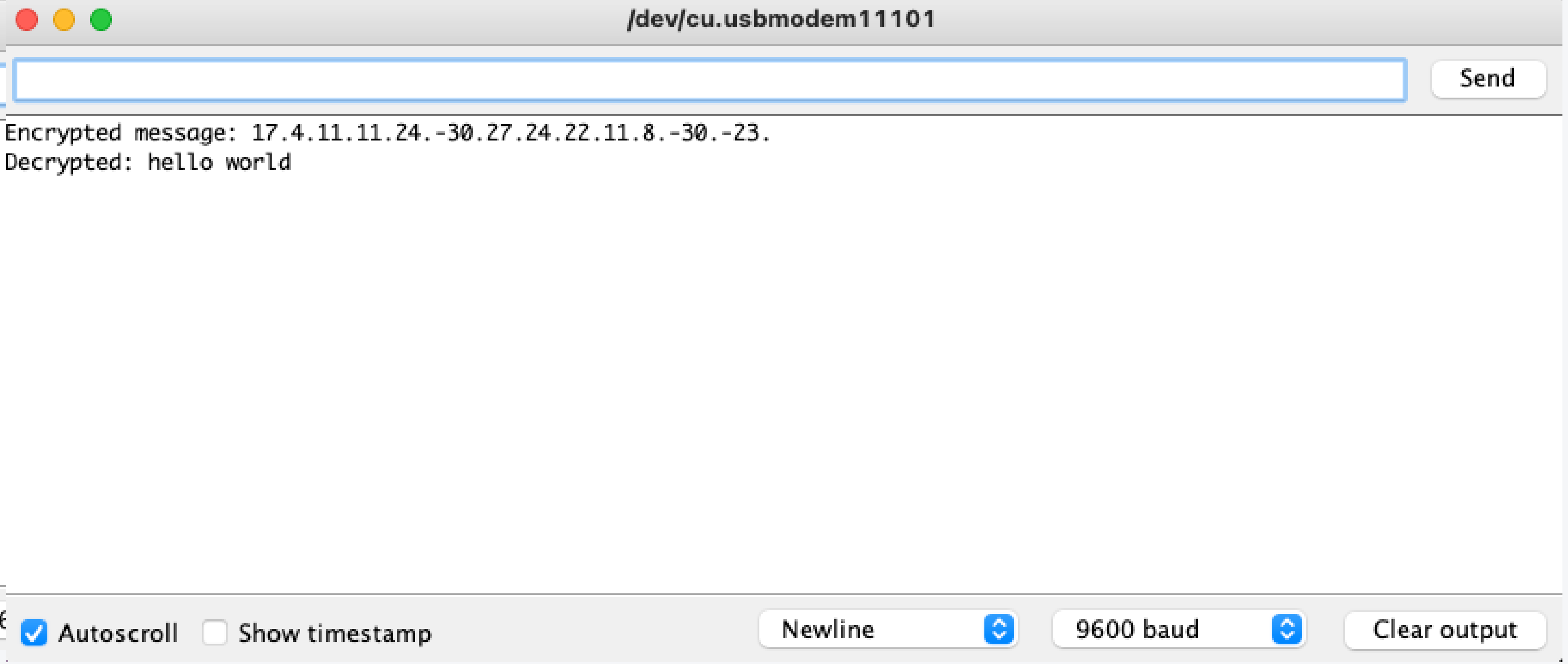
☑ Autoscroll ☐ Show timestamp Newline 9600 baud



/dev/cu.usbmodem11101

```
Encrypted message: 17.4.11.11.24.-30.27.24.22.11.8.-30.-23.
```

☑ Autoscroll ☐ Show timestamp Newline 96

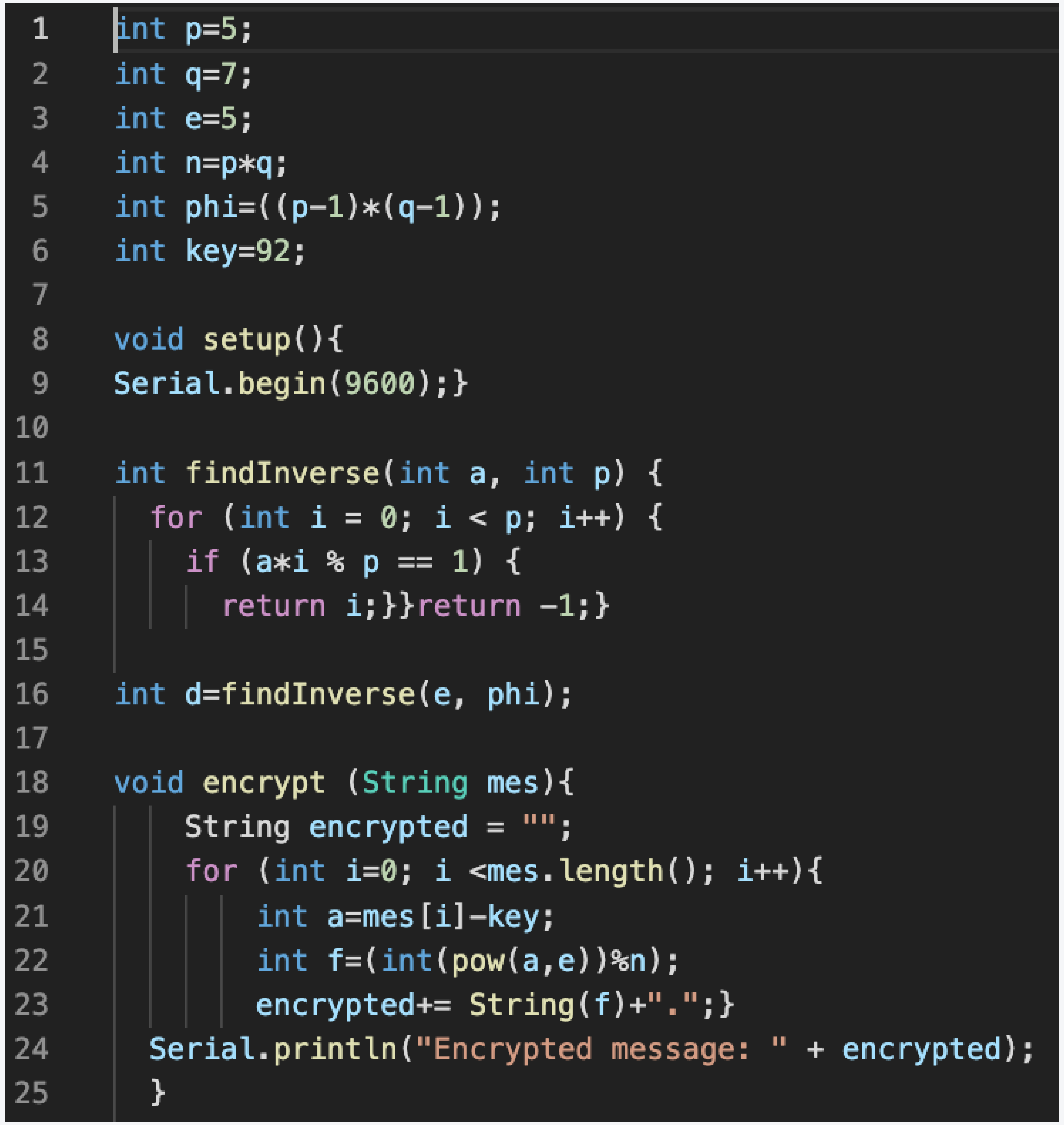


/dev/cu.usbmodem11101

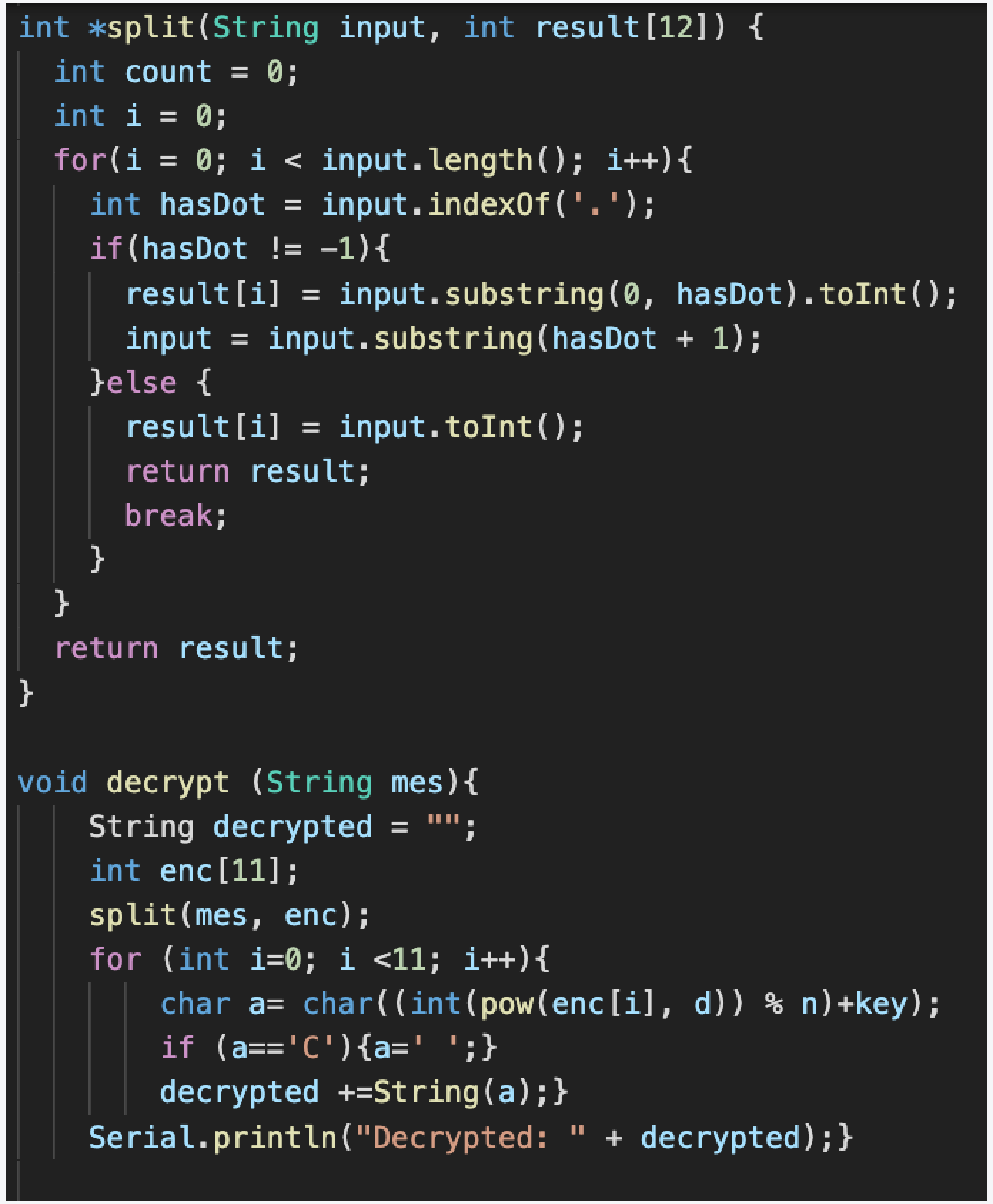
```
Encrypted message: 17.4.11.11.24.-30.27.24.22.11.8.-30.-23.
Decrypted: hello world
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud

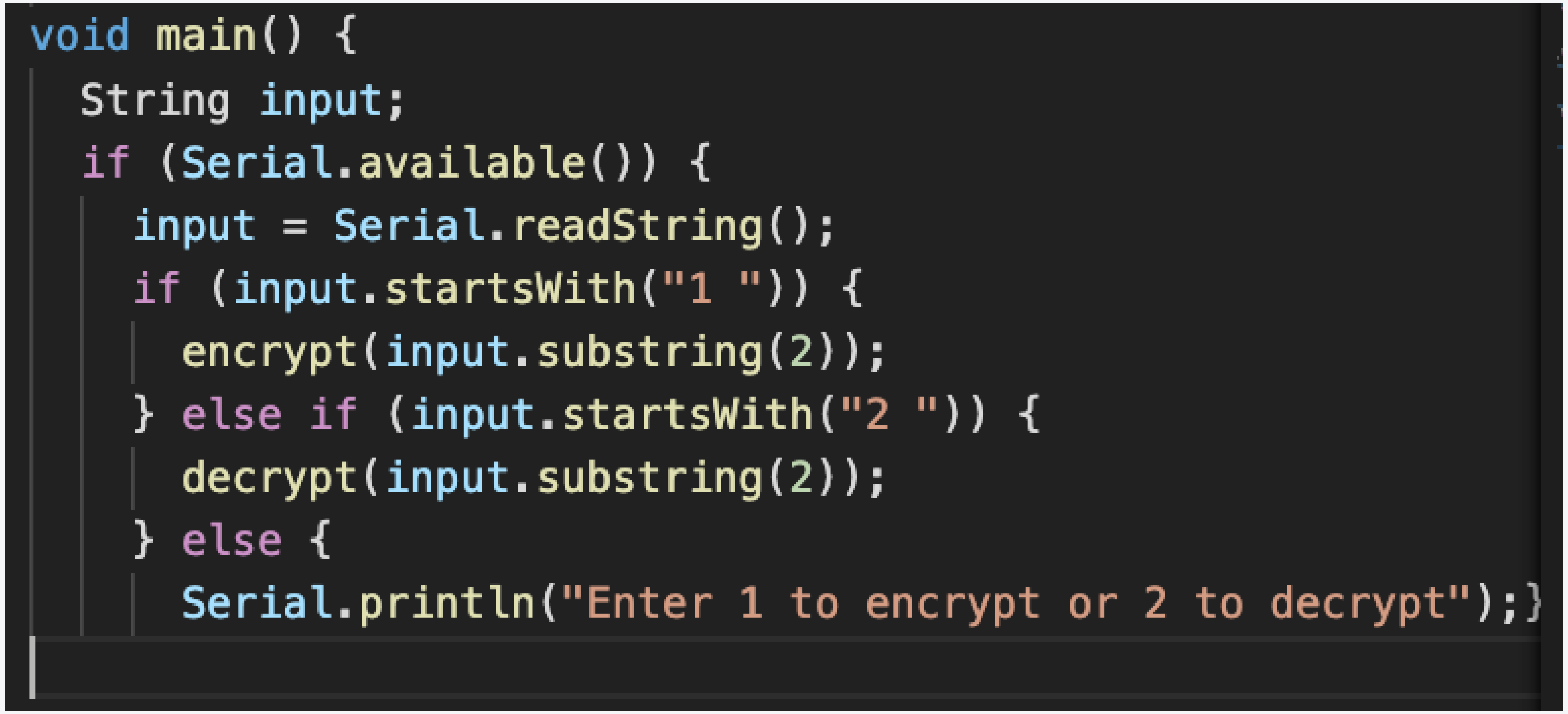
Annex



```
int p=5;
int q=7;
int e=5;
int n=p*q;
int phi=((p-1)*(q-1));
int key=92;

void setup(){
Serial.begin(9600);}

int findInverse(int a, int p) {
  for (int i = 0; i < p; i++) {
    if (a*i % p == 1) {
      return i;}}return -1;}

int d=findInverse(e, phi);

void encrypt (String mes){
    String encrypted = "";
    for (int i=0; i <mes.length(); i++){
        int a=mes[i]-key;
        int f=(int(pow(a,e))%n);
        encrypted+= String(f)+".";}
  Serial.println("Encrypted message: " + encrypted);
  }
```



```
int *split(String input, int result[12]) {
  int count = 0;
  int i = 0;
  for(i = 0; i < input.length(); i++){
    int hasDot = input.indexOf('.');
    if(hasDot != -1){
      result[i] = input.substring(0, hasDot).toInt();
      input = input.substring(hasDot + 1);
    }else {
      result[i] = input.toInt();
      return result;
      break;
    }
  }
  return result;
}

void decrypt (String mes){
    String decrypted = "";
    int enc[11];
    split(mes, enc);
    for (int i=0; i <11; i++){
        char a= char((int(pow(enc[i], d)) % n)+key);
        if (a=='C'){a=' ';}
        decrypted +=String(a);}
    Serial.println("Decrypted: " + decrypted);}
```



```
void main() {
  String input;
  if (Serial.available()) {
    input = Serial.readString();
    if (input.startsWith("1 ")) {
      encrypt(input.substring(2));
    } else if (input.startsWith("2 ")) {
      decrypt(input.substring(2));
    } else {
      Serial.println("Enter 1 to encrypt or 2 to decrypt");}
```

This code has been based on. http://koclab.cs.ucsb.edu/teaching/cren/project/2018/Adamczyk+Magnussen.pdf

The keys are based on the theory of the slides and with try and error. not anything else works for p q and e. but the key can be from 91 to 98.

The main keys are hardcoded and we run a table fro 12 charactes due to the fact we only want to encryp the hello world message. but it can be changed accordingly regarding the message we want to encrypt.