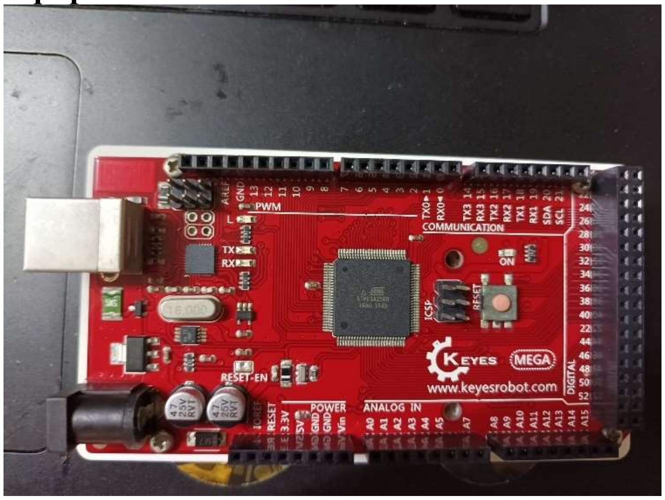
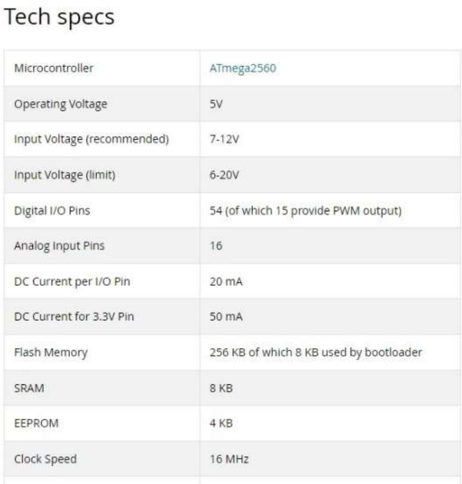
Future Hardware Testing Plans

To further validate the ACS Mini System's performance and security, the research project will incorporate two additional microcontroller boards for testing different encryption techniques. This expanded hardware testing phase aims to evaluate the system's data protection capabilities under various scenarios. The microcontrollers will be integrated into the existing setup, allowing for comprehensive testing of encryption algorithms and their impact on system performance. The testing results, including security assessments and performance metrics, will be documented and uploaded to the project's GitHub repository in the "documents" folder as a series of detailed reports. This iterative testing approach ensures the ACS Mini System meets the highest standards of data security and reliability, crucial for its successful deployment in construction environments.

Equipment: Arduino MEGA 2560

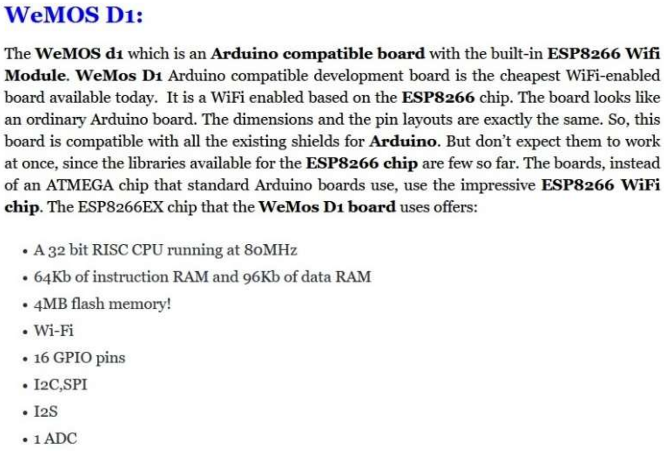




Tech specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Equipment: ESP8266 WeMos D1 R2



WeMOS D1:

The **WeMOS d1** which is an **Arduino compatible board** with the built-in **ESP8266 Wifi Module**. **WeMos D1** Arduino compatible development board is the cheapest WiFi-enabled board available today. It is a WiFi enabled based on the **ESP8266** chip. The board looks like an ordinary Arduino board. The dimensions and the pin layouts are exactly the same. So, this board is compatible with all the existing shields for **Arduino**. But don't expect them to work at once, since the libraries available for the **ESP8266 chip** are few so far. The boards, instead of an ATMEGA chip that standard Arduino boards use, use the impressive **ESP8266 WiFi chip**. The ESP8266EX chip that the **WeMos D1 board** uses offers:

- A 32 bit RISC CPU running at 80MHz
- 64Kb of instruction RAM and 96Kb of data RAM
- 4MB flash memory!
- Wi-Fi
- 16 GPIO pins
- I2C,SPI
- I2S
- 1 ADC

Test Result

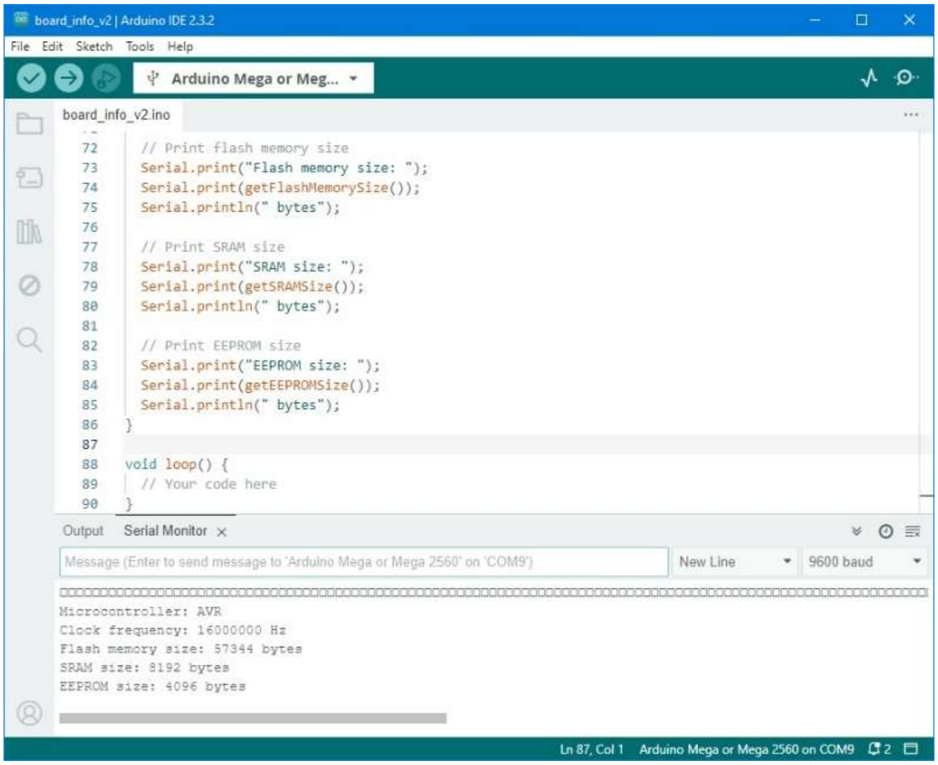
Arduino MEGA 2560

Encryption Algorithm	State Sizes (bytes)	Performance Tests (bytes per sec)	
		Encrypt	Decrypt
AES128	181	27962.53	14743.35
AES192	213	23267.88	12171.95
AES256	245	19923.04	10364.32
AESSmall128	34	23336.24	13492.51
AESSmall256	66	16852.51	9483.58
BLAKE2b	211	13482.85	
BLAKE2s	107	18305.26	
ChaCha20 128	132	19535.42	19533.18
ChaCha20 256	132	19535.42	19533.18
ChaCha12 128	132	30999.88	30994.18
ChaCha12 256	132	30999.88	30994.24
ChaCha8 128	132	43873.72	43862.29
ChaCha8 256	132	43873.60	43862.29
ChaCha+Poly1305	221	13110.32	13110.33
SHA-256	107	5985.56	
SHA3-256	205	16361.26	
SHAKE-256	206	16400.75	

WeMos D1 R2 ESP8266

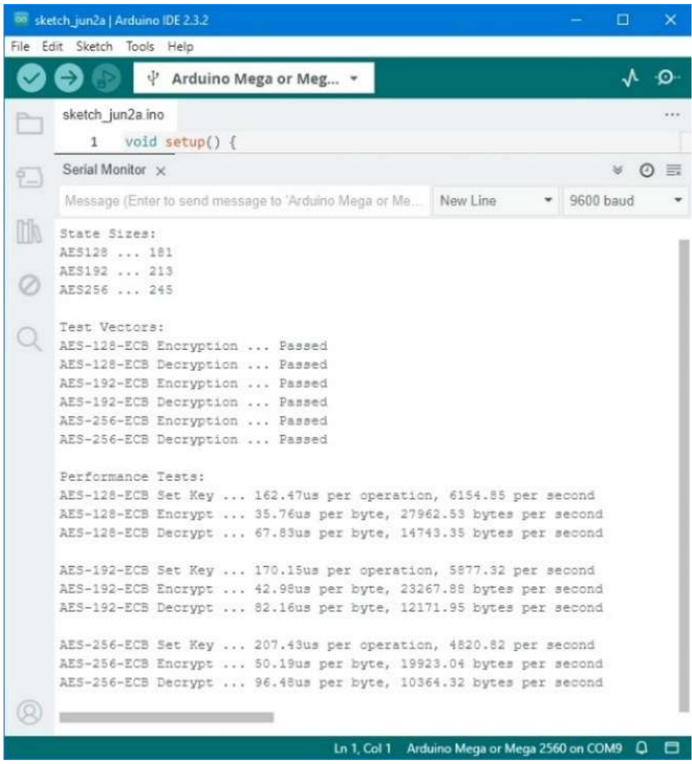
Encryption Algorithm	State Sizes (bytes)	Performance Tests (bytes per sec)	
		Encrypt	Decrypt
AES128	188	167929.28	112976.16
AES192	220	140069.51	93693.05
AES256	252	120139.78	80031.85
AESSmall128	36	130264.94	93754.87
AESSmall256	68	98547.29	68881.32
BLAKE2b	224	713987.57	
BLAKE2s	120	809885.67	
ChaCha20 128	130	1492989.95	1490694.80
ChaCha20 256	132	1493233.78	1490833.70
ChaCha12 128	132	1884736.58	1880748.77
ChaCha12 256	132	1884847.59	1880748.77
ChaCha8 128	132	2168903.35	2163697.22
ChaCha8 256	132	2169197.40	2163697.22
ChaCha+Poly1305	240	578745.57	578593.84
SHA-256	120	814166.50	
SHA3-256	224	259523.29	
SHAKE-256	232	258027.23	

Arduino MEGA 2560 Testing Result


board_info_v2 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
board_info_v2.ino
Output
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM9')
New Line
9600 baud
Ln 87, Col 1
Arduino Mega or Mega 2560 on COM9
2

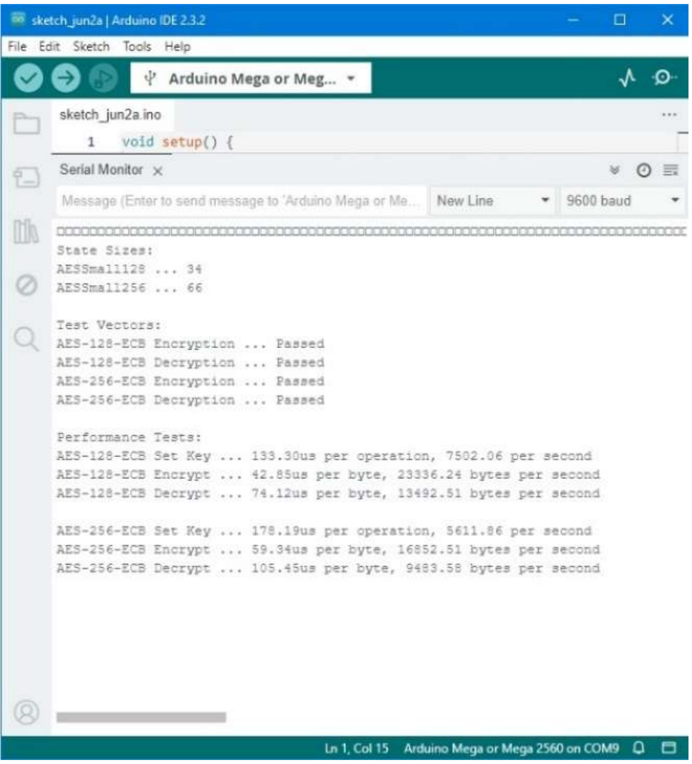
```
  // Print flash memory size
  Serial.print("Flash memory size: ");
  Serial.print(getFlashMemorySize());
  Serial.println(" bytes");

  // Print SRAM size
  Serial.print("SRAM size: ");
  Serial.print(getSRAMSize());
  Serial.println(" bytes");

  // Print EEPROM size
  Serial.print("EEPROM size: ");
  Serial.print(getEEPROMSize());
  Serial.println(" bytes");
}

void loop() {
  // Your code here
}
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Microcontroller: AVR
Clock frequency: 16000000 Hz
Flash memory size: 57344 bytes
SRAM size: 8192 bytes
EEPROM size: 4096 bytes
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
New Line
9600 baud
Ln 1, Col 1
Arduino Mega or Mega 2560 on COM9

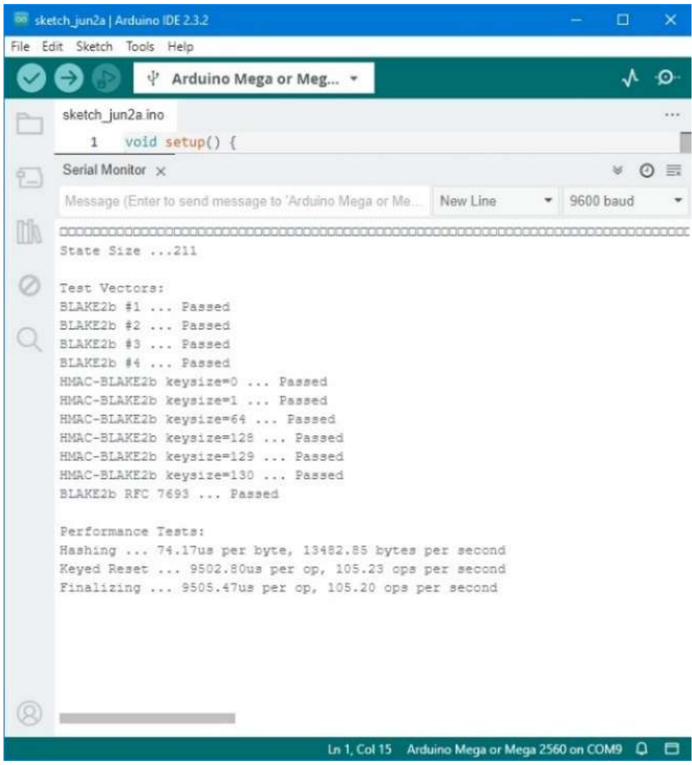
```
void setup() {
```

```
State Sizes:
AES128 ... 181
AES192 ... 213
AES256 ... 245

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-192-ECB Encryption ... Passed
AES-192-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 162.47us per operation, 6154.85 per second
AES-128-ECB Encrypt ... 35.76us per byte, 27962.53 bytes per second
AES-128-ECB Decrypt ... 67.83us per byte, 14743.35 bytes per second

AES-192-ECB Set Key ... 170.15us per operation, 5877.32 per second
AES-192-ECB Encrypt ... 42.98us per byte, 23267.88 bytes per second
AES-192-ECB Decrypt ... 82.16us per byte, 12171.95 bytes per second

AES-256-ECB Set Key ... 207.43us per operation, 4820.82 per second
AES-256-ECB Encrypt ... 50.19us per byte, 19923.04 bytes per second
AES-256-ECB Decrypt ... 96.48us per byte, 10364.32 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
New Line
9600 baud
Ln 1, Col 15
Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

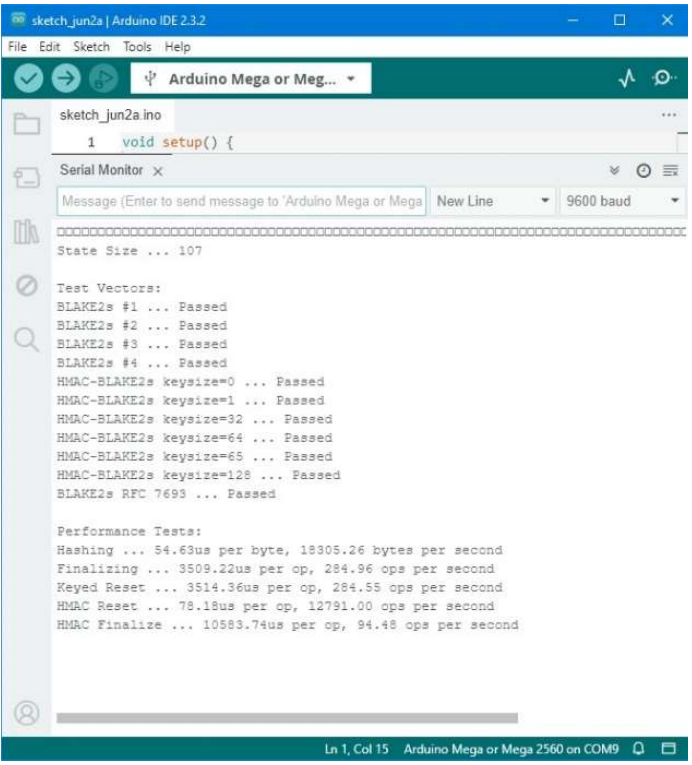
```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
State Sizes:
AESSmall128 ... 34
AESSmall256 ... 66

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 133.30us per operation, 7502.06 per second
AES-128-ECB Encrypt ... 42.85us per byte, 23336.24 bytes per second
AES-128-ECB Decrypt ... 74.12us per byte, 13492.51 bytes per second

AES-256-ECB Set Key ... 178.19us per operation, 5611.86 per second
AES-256-ECB Encrypt ... 59.34us per byte, 16852.51 bytes per second
AES-256-ECB Decrypt ... 105.45us per byte, 9483.58 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

New Line, 9600 baud

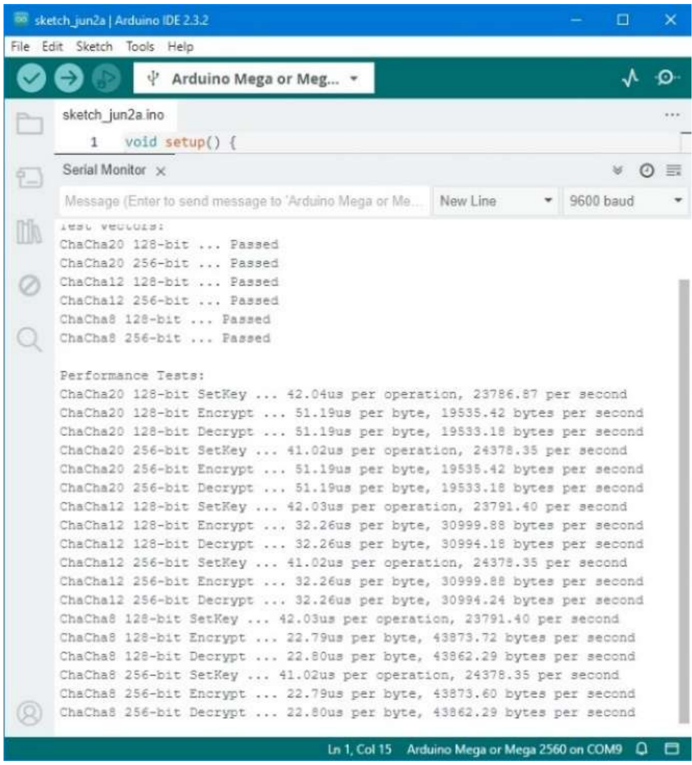
```
State Size ...211

Test Vectors:
BLAKE2b #1 ... Passed
BLAKE2b #2 ... Passed
BLAKE2b #3 ... Passed
BLAKE2b #4 ... Passed
HMAC-BLAKE2b keysize=0 ... Passed
HMAC-BLAKE2b keysize=1 ... Passed
HMAC-BLAKE2b keysize=64 ... Passed
HMAC-BLAKE2b keysize=128 ... Passed
HMAC-BLAKE2b keysize=129 ... Passed
HMAC-BLAKE2b keysize=130 ... Passed
BLAKE2b RFC 7693 ... Passed

Performance Tests:
Hashing ... 74.17us per byte, 13482.85 bytes per second
Keyed Reset ... 9502.80us per op, 105.23 ops per second
Finalizing ... 9505.47us per op, 105.20 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Mega
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

New Line, 9600 baud

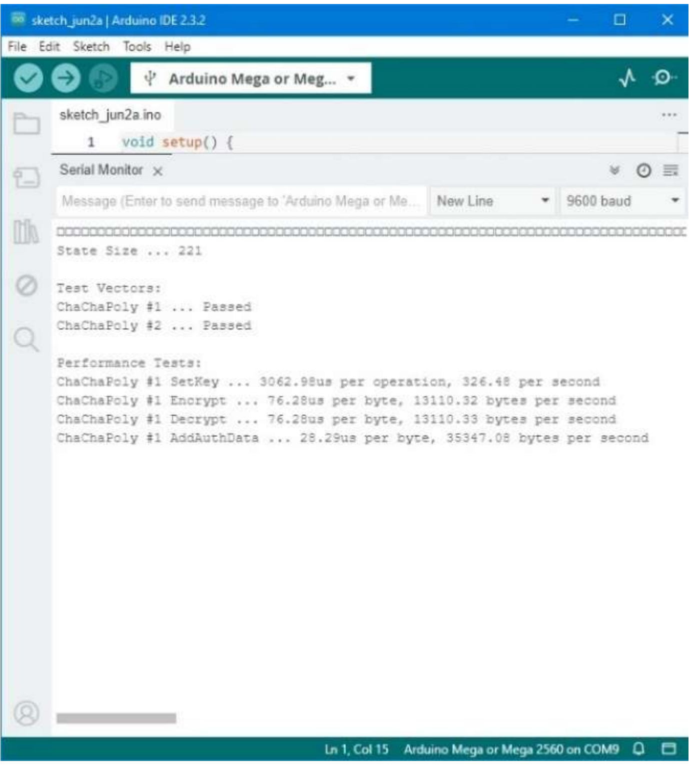
```
State Size ... 107

Test Vectors:
BLAKE2s #1 ... Passed
BLAKE2s #2 ... Passed
BLAKE2s #3 ... Passed
BLAKE2s #4 ... Passed
HMAC-BLAKE2s keysize=0 ... Passed
HMAC-BLAKE2s keysize=1 ... Passed
HMAC-BLAKE2s keysize=32 ... Passed
HMAC-BLAKE2s keysize=64 ... Passed
HMAC-BLAKE2s keysize=65 ... Passed
HMAC-BLAKE2s keysize=128 ... Passed
BLAKE2s RFC 7693 ... Passed

Performance Tests:
Hashing ... 54.63us per byte, 18305.26 bytes per second
Finalizing ... 3509.22us per op, 284.96 ops per second
Keyed Reset ... 3514.36us per op, 284.55 ops per second
HMAC Reset ... 78.18us per op, 12791.00 ops per second
HMAC Finalize ... 10583.74us per op, 94.48 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

New Line, 9600 baud

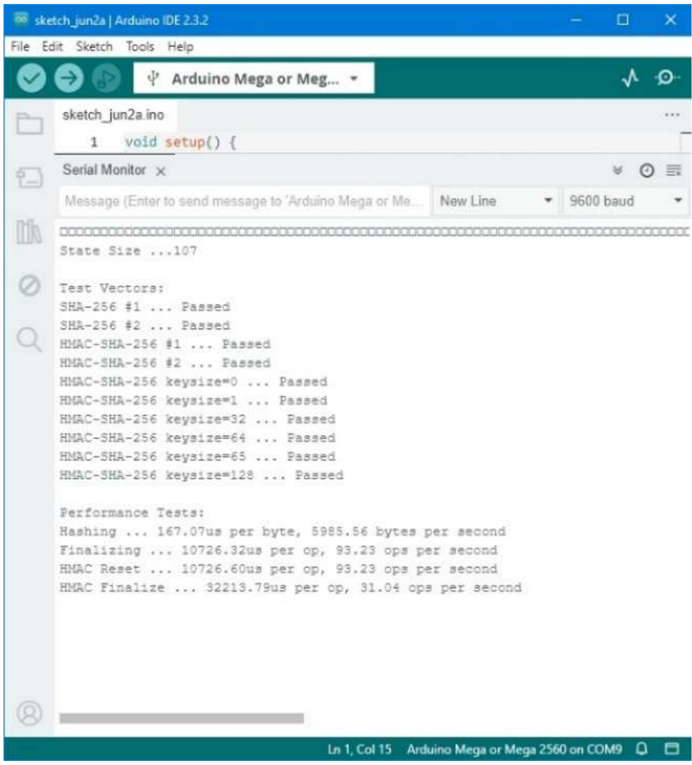
```
Test Vectors:
ChaCha20 128-bit ... Passed
ChaCha20 256-bit ... Passed
ChaCha12 128-bit ... Passed
ChaCha12 256-bit ... Passed
ChaCha8 128-bit ... Passed
ChaCha8 256-bit ... Passed

Performance Tests:
ChaCha20 128-bit SetKey ... 42.04us per operation, 23786.87 per second
ChaCha20 128-bit Encrypt ... 51.19us per byte, 19535.42 bytes per second
ChaCha20 128-bit Decrypt ... 51.19us per byte, 19533.18 bytes per second
ChaCha20 256-bit SetKey ... 41.02us per operation, 24378.35 per second
ChaCha20 256-bit Encrypt ... 51.19us per byte, 19535.42 bytes per second
ChaCha20 256-bit Decrypt ... 51.19us per byte, 19533.18 bytes per second
ChaCha12 128-bit SetKey ... 42.03us per operation, 23791.40 per second
ChaCha12 128-bit Encrypt ... 32.26us per byte, 30999.88 bytes per second
ChaCha12 128-bit Decrypt ... 32.26us per byte, 30994.18 bytes per second
ChaCha12 256-bit SetKey ... 41.02us per operation, 24378.35 per second
ChaCha12 256-bit Encrypt ... 32.26us per byte, 30999.88 bytes per second
ChaCha12 256-bit Decrypt ... 32.26us per byte, 30994.24 bytes per second
ChaCha8 128-bit SetKey ... 42.03us per operation, 23791.40 per second
ChaCha8 128-bit Encrypt ... 22.79us per byte, 43873.72 bytes per second
ChaCha8 128-bit Decrypt ... 22.80us per byte, 43862.29 bytes per second
ChaCha8 256-bit SetKey ... 41.02us per operation, 24378.35 per second
ChaCha8 256-bit Encrypt ... 22.79us per byte, 43873.60 bytes per second
ChaCha8 256-bit Decrypt ... 22.80us per byte, 43862.29 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

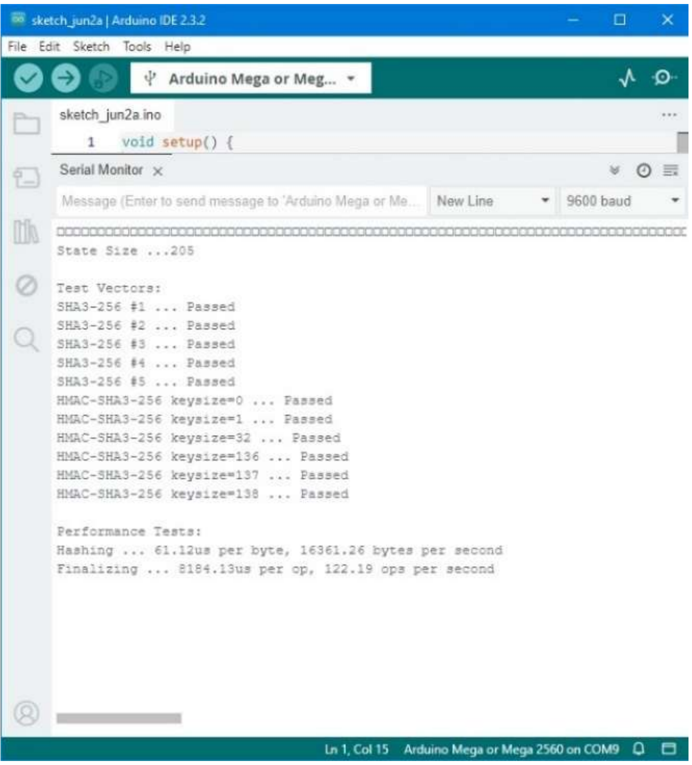
New Line, 9600 baud

```
State Size ... 221

Test Vectors:
ChaChaPoly #1 ... Passed
ChaChaPoly #2 ... Passed

Performance Tests:
ChaChaPoly #1 SetKey ... 3062.98us per operation, 326.48 per second
ChaChaPoly #1 Encrypt ... 76.28us per byte, 13110.32 bytes per second
ChaChaPoly #1 Decrypt ... 76.28us per byte, 13110.33 bytes per second
ChaChaPoly #1 AddAuthData ... 28.29us per byte, 35347.08 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
New Line
9600 baud
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

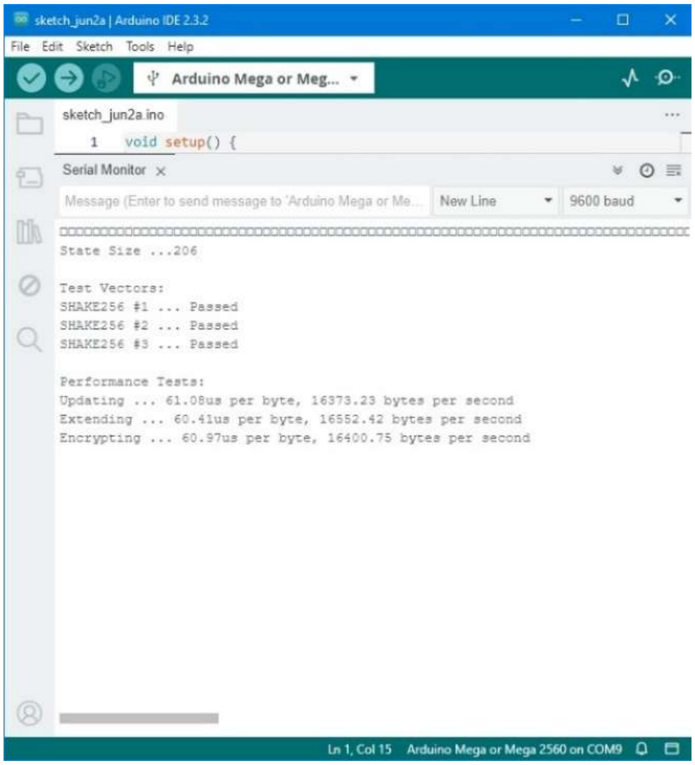
```
void setup() {
```

```
State Size ...107

Test Vectors:
SHA-256 #1 ... Passed
SHA-256 #2 ... Passed
HMAC-SHA-256 #1 ... Passed
HMAC-SHA-256 #2 ... Passed
HMAC-SHA-256 keysize=0 ... Passed
HMAC-SHA-256 keysize=1 ... Passed
HMAC-SHA-256 keysize=32 ... Passed
HMAC-SHA-256 keysize=64 ... Passed
HMAC-SHA-256 keysize=65 ... Passed
HMAC-SHA-256 keysize=128 ... Passed

Performance Tests:
Hashing ... 167.07us per byte, 5985.56 bytes per second
Finalizing ... 10726.32us per op, 93.23 ops per second
HMAC Reset ... 10726.60us per op, 93.23 ops per second
HMAC Finalize ... 32213.79us per op, 31.04 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
New Line
9600 baud
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

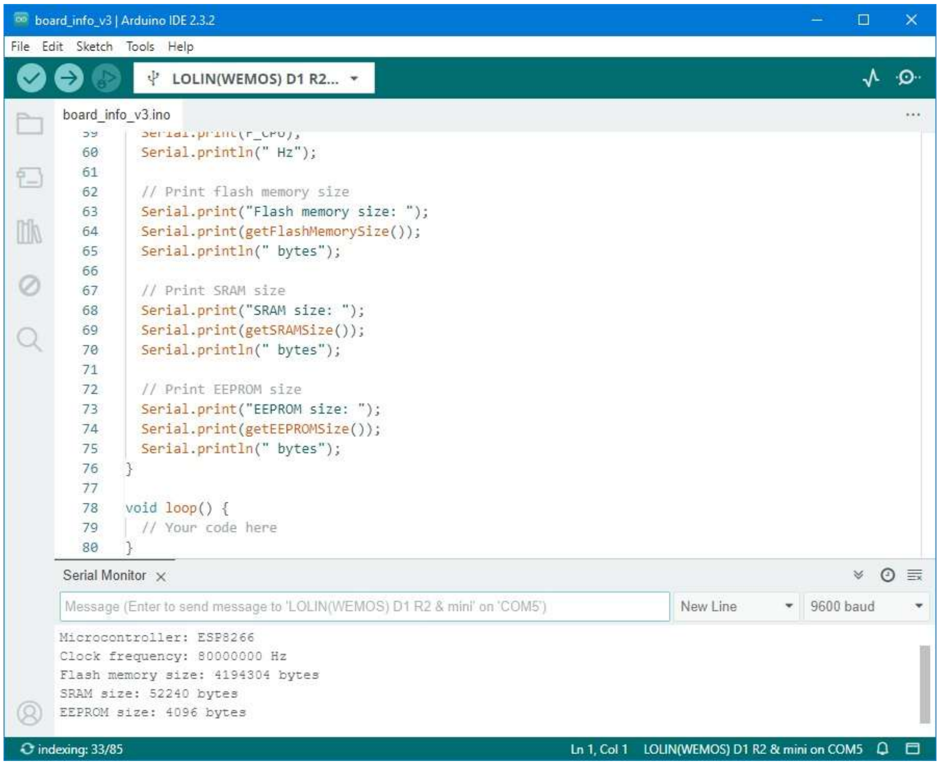
```
State Size ...205

Test Vectors:
SHA3-256 #1 ... Passed
SHA3-256 #2 ... Passed
SHA3-256 #3 ... Passed
SHA3-256 #4 ... Passed
SHA3-256 #5 ... Passed
HMAC-SHA3-256 keysize=0 ... Passed
HMAC-SHA3-256 keysize=1 ... Passed
HMAC-SHA3-256 keysize=32 ... Passed
HMAC-SHA3-256 keysize=136 ... Passed
HMAC-SHA3-256 keysize=137 ... Passed
HMAC-SHA3-256 keysize=138 ... Passed

Performance Tests:
Hashing ... 61.12us per byte, 16361.26 bytes per second
Finalizing ... 8184.13us per op, 122.19 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Mega or Meg...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'Arduino Mega or Me...
New Line
9600 baud
Ln 1, Col 15 Arduino Mega or Mega 2560 on COM9

```
void setup() {
```

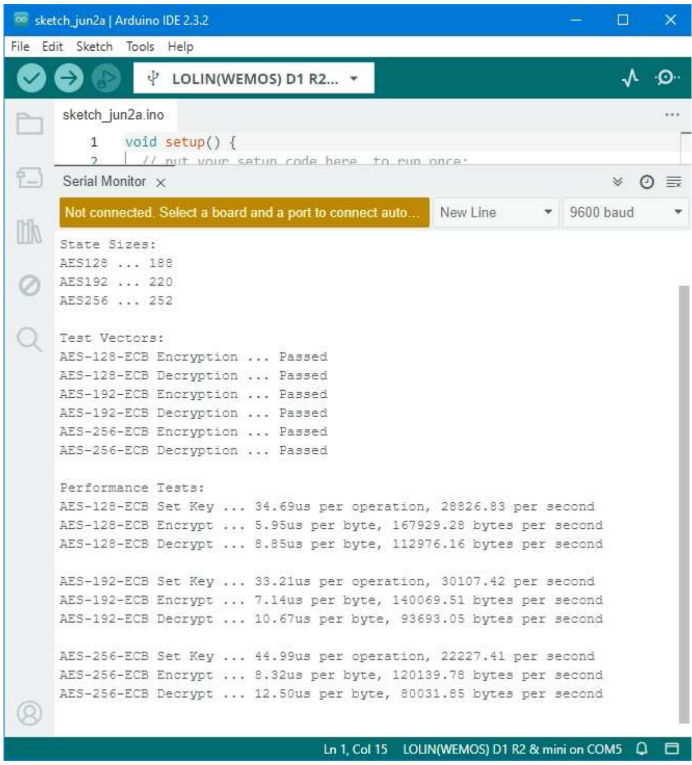
```
State Size ...206

Test Vectors:
SHAKE256 #1 ... Passed
SHAKE256 #2 ... Passed
SHAKE256 #3 ... Passed

Performance Tests:
Updating ... 61.08us per byte, 16373.23 bytes per second
Extending ... 60.41us per byte, 16552.42 bytes per second
Encrypting ... 60.97us per byte, 16400.75 bytes per second
```

WeMos D1 R2 ESP8266 Testing Result


board_info_v3 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
board_info_v3.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 R2 & mini' on 'COM5')
New Line
9600 baud
indexing: 33/85
Ln 1, Col 1 LOLIN(WEMOS) D1 R2 & mini on COM5

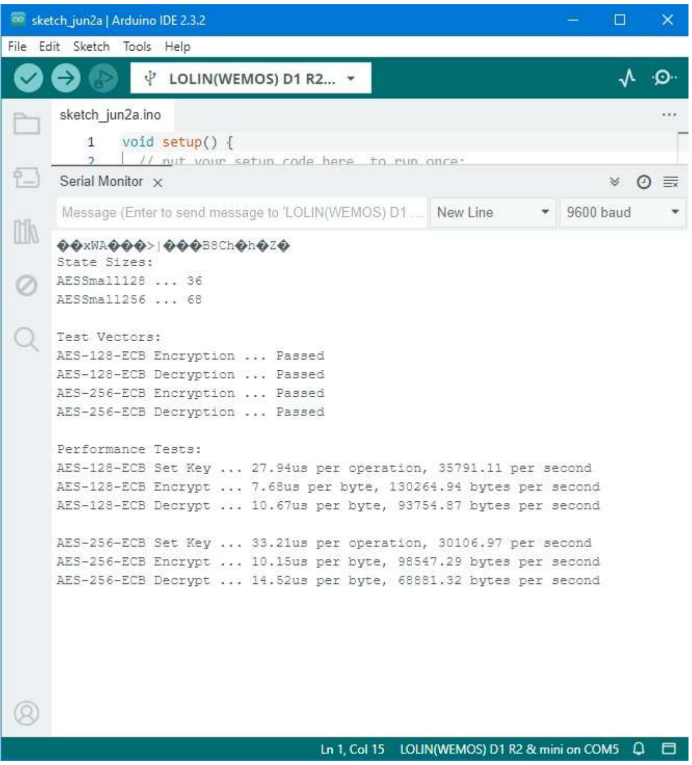
```
  Serial.print(F_CPU);
  Serial.println(" Hz");

  // Print flash memory size
  Serial.print("Flash memory size: ");
  Serial.print(getFlashMemorySize());
  Serial.println(" bytes");

  // Print SRAM size
  Serial.print("SRAM size: ");
  Serial.print(getSRAMSize());
  Serial.println(" bytes");

  // Print EEPROM size
  Serial.print("EEPROM size: ");
  Serial.print(getEEPROMSize());
  Serial.println(" bytes");
}

void loop() {
  // Your code here
}
```

```
Microcontroller: ESP8266
Clock frequency: 80000000 Hz
Flash memory size: 4194304 bytes
SRAM size: 52240 bytes
EEPROM size: 4096 bytes
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Not connected. Select a board and a port to connect auto...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

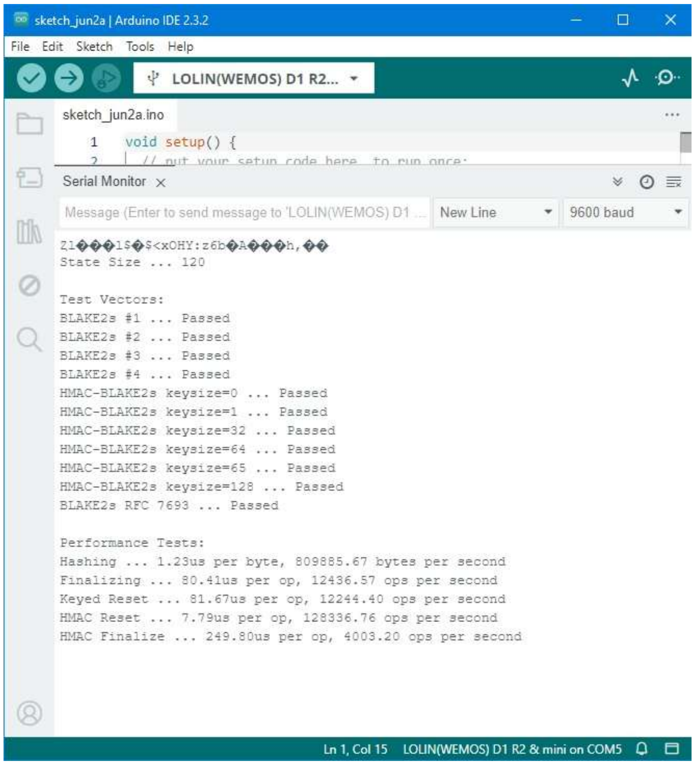
```
void setup() {
  // put your setup code here, to run once:
```

```
State Sizes:
AES128 ... 188
AES192 ... 220
AES256 ... 252

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-192-ECB Encryption ... Passed
AES-192-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 34.69us per operation, 28826.83 per second
AES-128-ECB Encrypt ... 5.95us per byte, 167929.28 bytes per second
AES-128-ECB Decrypt ... 8.85us per byte, 112976.16 bytes per second

AES-192-ECB Set Key ... 33.21us per operation, 30107.42 per second
AES-192-ECB Encrypt ... 7.14us per byte, 140069.51 bytes per second
AES-192-ECB Decrypt ... 10.67us per byte, 93693.05 bytes per second

AES-256-ECB Set Key ... 44.99us per operation, 22227.41 per second
AES-256-ECB Encrypt ... 8.32us per byte, 120139.78 bytes per second
AES-256-ECB Decrypt ... 12.50us per byte, 80031.85 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

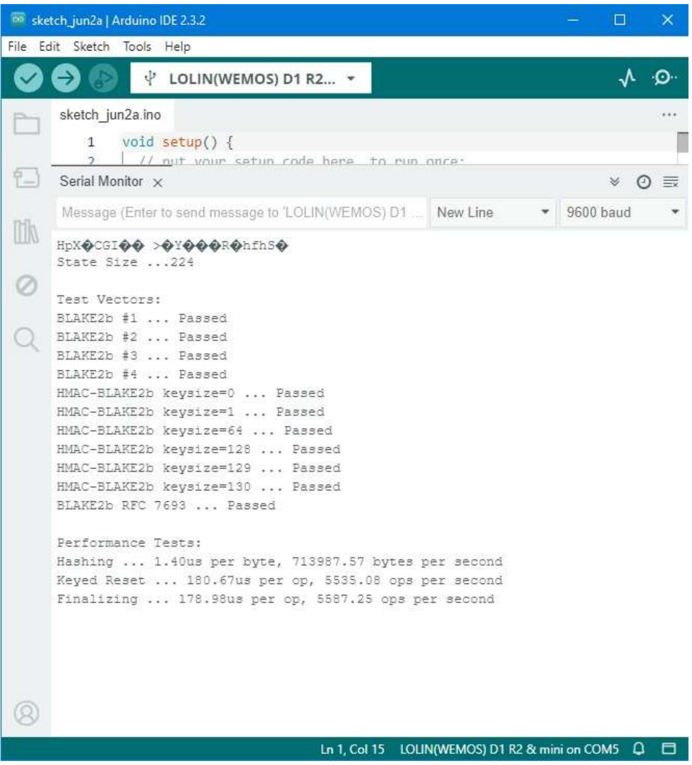
```
void setup() {
  // put your setup code here, to run once:
```

```
��xWA���>|���B8Ch�h�Z�
State Sizes:
AESSmall128 ... 36
AESSmall256 ... 68

Test Vectors:
AES-128-ECB Encryption ... Passed
AES-128-ECB Decryption ... Passed
AES-256-ECB Encryption ... Passed
AES-256-ECB Decryption ... Passed

Performance Tests:
AES-128-ECB Set Key ... 27.94us per operation, 35791.11 per second
AES-128-ECB Encrypt ... 7.68us per byte, 130264.94 bytes per second
AES-128-ECB Decrypt ... 10.67us per byte, 93754.87 bytes per second

AES-256-ECB Set Key ... 33.21us per operation, 30106.97 per second
AES-256-ECB Encrypt ... 10.15us per byte, 98547.29 bytes per second
AES-256-ECB Decrypt ... 14.52us per byte, 68881.32 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

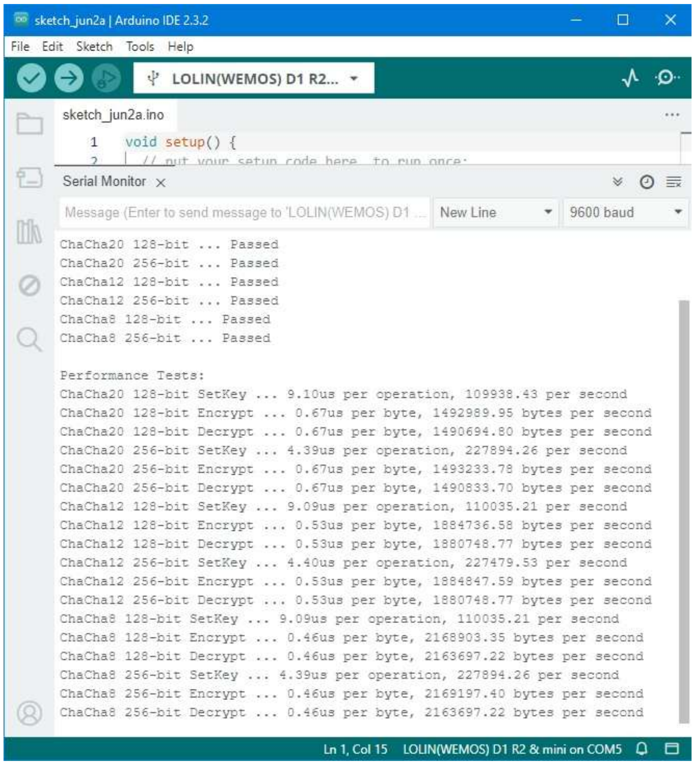
```
void setup() {
  // put your setup code here, to run once:
```

```
Z1���1$�$<xOHY:z6b�A���h,��
State Size ... 120

Test Vectors:
BLAKE2s #1 ... Passed
BLAKE2s #2 ... Passed
BLAKE2s #3 ... Passed
BLAKE2s #4 ... Passed
HMAC-BLAKE2s keysize=0 ... Passed
HMAC-BLAKE2s keysize=1 ... Passed
HMAC-BLAKE2s keysize=32 ... Passed
HMAC-BLAKE2s keysize=64 ... Passed
HMAC-BLAKE2s keysize=65 ... Passed
HMAC-BLAKE2s keysize=128 ... Passed
BLAKE2s RFC 7693 ... Passed

Performance Tests:
Hashing ... 1.23us per byte, 809885.67 bytes per second
Finalizing ... 80.41us per op, 12436.57 ops per second
Keyed Reset ... 81.67us per op, 12244.40 ops per second
HMAC Reset ... 7.79us per op, 128336.76 ops per second
HMAC Finalize ... 249.80us per op, 4003.20 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

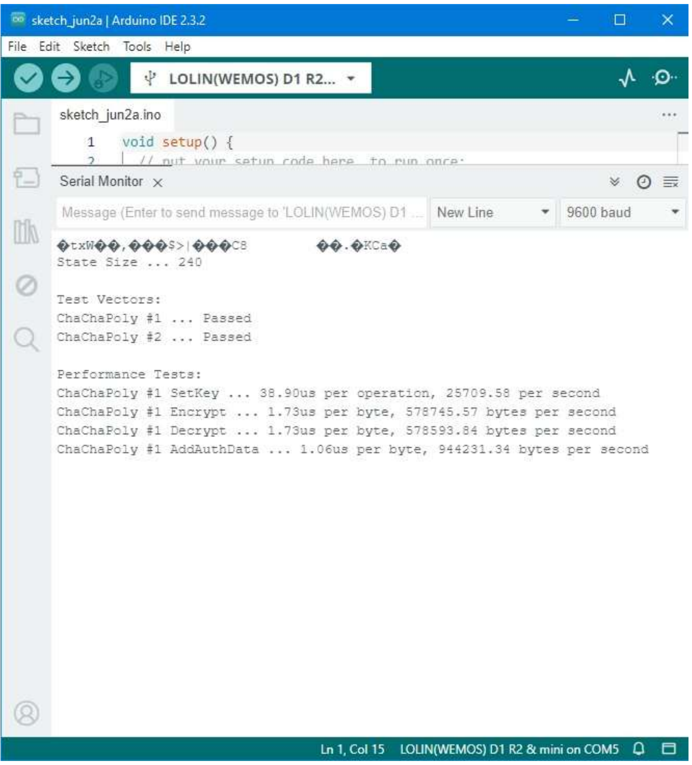
```
void setup() {
  // put your setup code here, to run once:
```

```
HpX�CGI�� >�Y���R�hfhS�
State Size ...224

Test Vectors:
BLAKE2b #1 ... Passed
BLAKE2b #2 ... Passed
BLAKE2b #3 ... Passed
BLAKE2b #4 ... Passed
HMAC-BLAKE2b keysize=0 ... Passed
HMAC-BLAKE2b keysize=1 ... Passed
HMAC-BLAKE2b keysize=64 ... Passed
HMAC-BLAKE2b keysize=128 ... Passed
HMAC-BLAKE2b keysize=129 ... Passed
HMAC-BLAKE2b keysize=130 ... Passed
BLAKE2b RFC 7693 ... Passed

Performance Tests:
Hashing ... 1.40us per byte, 713987.57 bytes per second
Keyed Reset ... 180.67us per op, 5535.08 ops per second
Finalizing ... 178.98us per op, 5587.25 ops per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

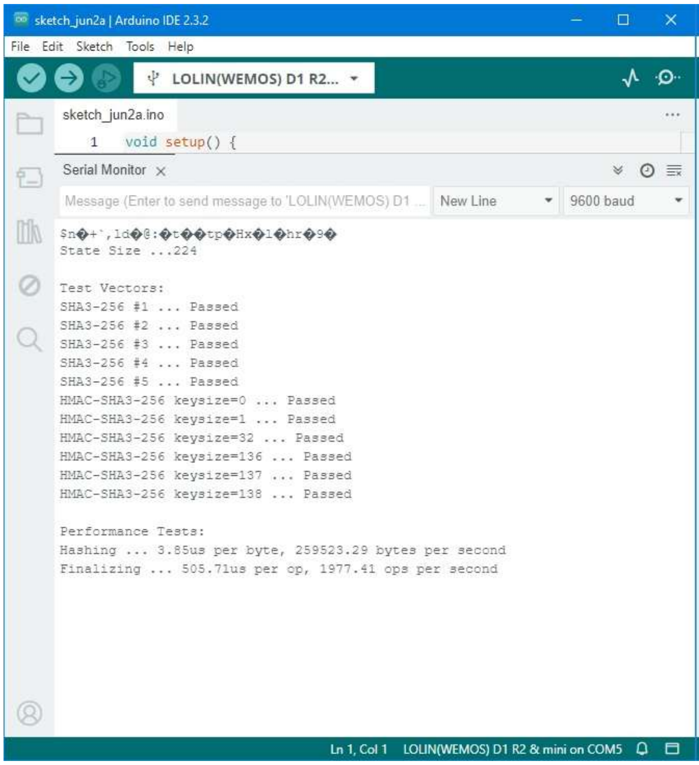
```
void setup() {
  // put your setup code here, to run once:
```

```
ChaCha20 128-bit ... Passed
ChaCha20 256-bit ... Passed
ChaCha12 128-bit ... Passed
ChaCha12 256-bit ... Passed
ChaCha8 128-bit ... Passed
ChaCha8 256-bit ... Passed

Performance Tests:
ChaCha20 128-bit SetKey ... 9.10us per operation, 109938.43 per second
ChaCha20 128-bit Encrypt ... 0.67us per byte, 1492989.95 bytes per second
ChaCha20 128-bit Decrypt ... 0.67us per byte, 1490694.80 bytes per second
ChaCha20 256-bit SetKey ... 4.39us per operation, 227894.26 per second
ChaCha20 256-bit Encrypt ... 0.67us per byte, 1493233.78 bytes per second
ChaCha20 256-bit Decrypt ... 0.67us per byte, 1490833.70 bytes per second
ChaCha12 128-bit SetKey ... 9.09us per operation, 110035.21 per second
ChaCha12 128-bit Encrypt ... 0.53us per byte, 1884736.58 bytes per second
ChaCha12 128-bit Decrypt ... 0.53us per byte, 1880748.77 bytes per second
ChaCha12 256-bit SetKey ... 4.40us per operation, 227479.53 per second
ChaCha12 256-bit Encrypt ... 0.53us per byte, 1884847.59 bytes per second
ChaCha12 256-bit Decrypt ... 0.53us per byte, 1880748.77 bytes per second
ChaCha8 128-bit SetKey ... 9.09us per operation, 110035.21 per second
ChaCha8 128-bit Encrypt ... 0.46us per byte, 2168903.35 bytes per second
ChaCha8 128-bit Decrypt ... 0.46us per byte, 2163697.22 bytes per second
ChaCha8 256-bit SetKey ... 4.39us per operation, 227894.26 per second
ChaCha8 256-bit Encrypt ... 0.46us per byte, 2169197.40 bytes per second
ChaCha8 256-bit Decrypt ... 0.46us per byte, 2163697.22 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud
Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5

```
void setup() {
  // put your setup code here, to run once:
```

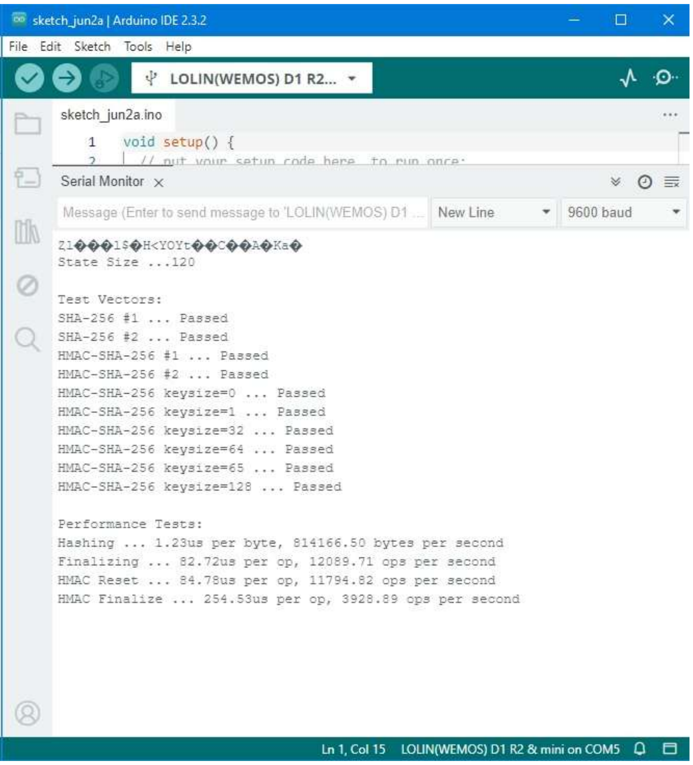
```
�txW��,���$>|���C8        ��.�KCa�
State Size ... 240

Test Vectors:
ChaChaPoly #1 ... Passed
ChaChaPoly #2 ... Passed

Performance Tests:
ChaChaPoly #1 SetKey ... 38.90us per operation, 25709.58 per second
ChaChaPoly #1 Encrypt ... 1.73us per byte, 578745.57 bytes per second
ChaChaPoly #1 Decrypt ... 1.73us per byte, 578593.84 bytes per second
ChaChaPoly #1 AddAuthData ... 1.06us per byte, 944231.34 bytes per second
```


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud

```
void setup() {
```

```
$n�+`,ld�@:�t��tp�Hx�l�hr�9�
State Size ...224

Test Vectors:
SHA3-256 #1 ... Passed
SHA3-256 #2 ... Passed
SHA3-256 #3 ... Passed
SHA3-256 #4 ... Passed
SHA3-256 #5 ... Passed
HMAC-SHA3-256 keysize=0 ... Passed
HMAC-SHA3-256 keysize=1 ... Passed
HMAC-SHA3-256 keysize=32 ... Passed
HMAC-SHA3-256 keysize=136 ... Passed
HMAC-SHA3-256 keysize=137 ... Passed
HMAC-SHA3-256 keysize=138 ... Passed

Performance Tests:
Hashing ... 3.85us per byte, 259523.29 bytes per second
Finalizing ... 505.71us per op, 1977.41 ops per second
```

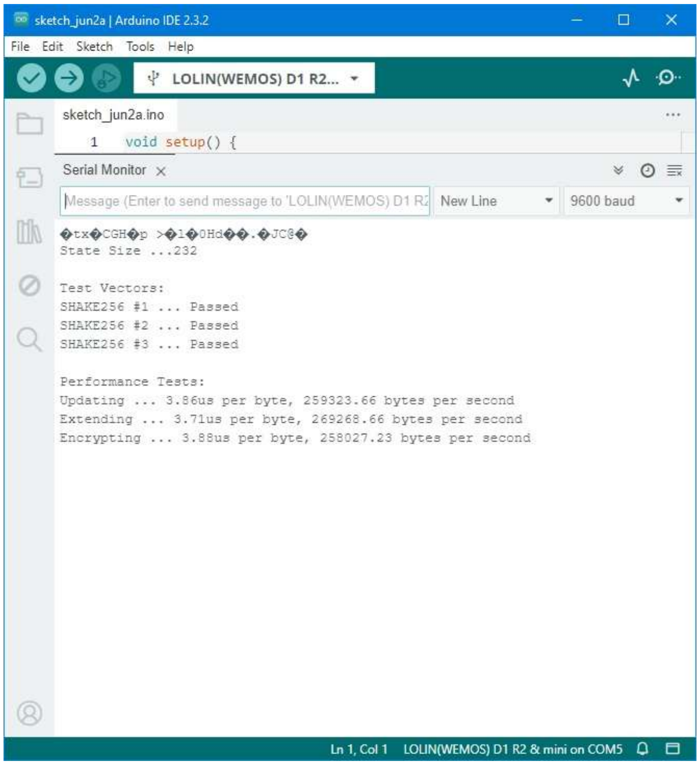
Ln 1, Col 1 LOLIN(WEMOS) D1 R2 & mini on COM5


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 ...
New Line
9600 baud

```
void setup() {
  // put your setup code here, to run once:
```

```
Z1���1$�H<YOYt��C��A�Ka�
State Size ...120

Test Vectors:
SHA-256 #1 ... Passed
SHA-256 #2 ... Passed
HMAC-SHA-256 #1 ... Passed
HMAC-SHA-256 #2 ... Passed
HMAC-SHA-256 keysize=0 ... Passed
HMAC-SHA-256 keysize=1 ... Passed
HMAC-SHA-256 keysize=32 ... Passed
HMAC-SHA-256 keysize=64 ... Passed
HMAC-SHA-256 keysize=65 ... Passed
HMAC-SHA-256 keysize=128 ... Passed

Performance Tests:
Hashing ... 1.23us per byte, 814166.50 bytes per second
Finalizing ... 82.72us per op, 12089.71 ops per second
HMAC Reset ... 84.78us per op, 11794.82 ops per second
HMAC Finalize ... 254.53us per op, 3928.89 ops per second
```

Ln 1, Col 15 LOLIN(WEMOS) D1 R2 & mini on COM5


sketch_jun2a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2...
sketch_jun2a.ino
Serial Monitor
Message (Enter to send message to 'LOLIN(WEMOS) D1 R2
New Line
9600 baud

```
void setup() {
```

```
�tx�CGH�p >�1�0Hd��.�JC@�
State Size ...232

Test Vectors:
SHAKE256 #1 ... Passed
SHAKE256 #2 ... Passed
SHAKE256 #3 ... Passed

Performance Tests:
Updating ... 3.86us per byte, 259323.66 bytes per second
Extending ... 3.71us per byte, 269268.66 bytes per second
Encrypting ... 3.88us per byte, 258027.23 bytes per second
```

Ln 1, Col 1 LOLIN(WEMOS) D1 R2 & mini on COM5

Additional Conclusion and Recommendations

The additional performance evaluation of encryption algorithms on resource-constrained devices has provided valuable insights into their suitability for secure data monitoring in the construction industry. The results highlight the trade-offs between security strength, performance, and resource usage, which are crucial considerations for developing an affordable and efficient ACS Mini System tailored for construction SMEs.

Among the tested algorithms, ChaCha12 and ChaCha8 demonstrated remarkably high encryption and decryption performance, making them promising candidates for real-time data encryption and decryption. The AES algorithms (AES128, AES192, and AES256) also exhibited good performance, with higher throughput rates on the more powerful WeMos D1 R2 platform. Other algorithms as BLAKE2b, BLAKE2s, SHA-256, SHA3-256, and SHAKE-256 showed impressive performance, suitable for scenarios prioritizing data integrity and authentication over confidentiality. The ChaCha+Poly1305 algorithm, providing authenticated encryption, could be valuable for secure communication channels where data authenticity is crucial.

Recommendations:

1. Adopt ChaCha12 (128-bit or 256-bit) as the primary encryption algorithm for real-time data encryption and decryption within the ACS Mini System, leveraging its high throughput rates and security strength.

2. Implement ChaCha+Poly1305 for secure communication channels, ensuring both data confidentiality and integrity.

3. Leverage other algorithms like BLAKE2b, BLAKE2s, SHA-256, SHA3-256, and SHAKE-256 for specific use cases prioritizing data integrity and authentication.

4. Explore hardware acceleration techniques and code optimizations to further improve encryption algorithm performance on resource-constrained devices.

5. Continuously monitor the security landscape and periodically re-evaluate the selected algorithms to ensure they remain secure and efficient.

6. Collaborate with industry partners, regulatory bodies, and construction SMEs to establish best practices and standards for secure data monitoring, ensuring interoperability and widespread adoption.

By following these recommendations, the ACS Mini System project can enhance the security and efficiency of data monitoring for construction SMEs, addressing resource constraints while ensuring compliance with industry standards and regulations.