This is an AES implementation that uses only 8-bit byte operations on the cipher state (there are options to use 32-bit types if available).

The s-box and inverse s-box were retained as tables (0.5kB PROGMEM) but all the other transformations are coded to save table space. Many efficiency improvments to the routines mix\_sub\_columns() and inv\_mix\_sub\_columns() (mainly common sub-expression elimination).

Only the routines with precalculated subkey schedule are retained (together with set\_key() - this does however mean each AES object takes 240 bytes of RAM, alas)

The CBC routines side-effect the iv argument (so that successive calls work together correctly).

All the encryption and decryption routines work with plain == cipher for in-place encryption, note

At the moment only 128bit keys are supported, the blocksize is also fixed at 128bit. This means that the key array and possible iv array should contain exactly 16 bytes (uint8\_t or byte). Moreover the amount of bytes to encrypt should be mod 16. (this means you have to take care of padding yourself).

The library supports 3 kinds of operations.

1. single block encryption/decryption

* multiple block encryption/decryption using CBC (single call)
* multiple block encryption/decryption using CBC (multiple calls)

The single block enc/decryption are the following methods:

void aes128\_enc\_single(const uint8\_t\* key, void\* data);

void aes128\_dec\_single(const uint8\_t\* key, void\* data);

Usage example:

Serial.begin(57600);

uint8\_t key[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};

char data[] = "0123456789012345"; //16 chars == 16 bytes

aes128\_enc\_single(key, data);

Serial.print("encrypted:");

Serial.println(data);

aes128\_dec\_single(key, data);

Serial.print("decrypted:");

Serial.println(data);

Usage example for AES256:

Serial.begin(57600);

uint8\_t key[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31};

char data[] = "0123456789012345";

aes256\_enc\_single(key, data);

Serial.print("encrypted:");

Serial.println(data);

aes256\_dec\_single(key, data);

Serial.print("decrypted:");

Serial.println(data);