begin()

Description

Initializes the network parameters of PHPoC Shield for Arduino.

Syntax

Phpoc.begin()

Phpoc.begin(debug_flag)

Parameters

debug_flag - flags for debugging

Debug Flags	Descriptions
PF_LOG_SPI	debugging flag for SPI communication
PF_LOG_NET	debugging flag for network communication
PF_LOG_APP	debugging flag for applications such as sending an E-mail

Returns

1 - on success

0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

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

    if(Phpoc.begin() != 0)
        Serial.println("Success");
    else
        Serial.println("Fail");
}

void loop()
{
}
```

localIP()

Description

This function is used to obtains an IP address of the PHPoC shield for Arduino.

Syntax

Phpoc.localIP()

Parameters

none

Returns

Returns a string represents an IP address. (e.g. 192.168.0.1)

Example

```
#include <SPI.h>
#include <Phpoc.h>

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

  if(Phpoc.begin() == 0)
  {
    Serial.println("Failed to initialize Network");
    for(;;)
      ;
  }
  Serial.println(Phpoc.localIP());
}

void loop()
{
}
```

beginIP6()

Description

Enables IPv6 feature.

Syntax

Phpoc.beginIP6()

Parameters

none

Returns

1 - on success
0 - on failure

example

```
#include <SPI.h>
#include <Phpoc.h>

void setup(){
  Serial.begin(9600);
  Phpoc.begin();
  if(Phpoc.beginIP6() != 0)
    Serial.println("Success");
  else
    Serial.println("Fail");
}

void loop(){
}
```

localIP6()

Description

This function is used to obtains a link local IPv6 address of the PHPoC shield for Arduino.

Syntax

Phpoc.localIP6()

Parameters

none

Returns

Returns a string represents a link local IPv6 address. (e.g. fe80:db8:131f::140b)

Example

```
#include <SPI.h>
#include <Phpoc.h>

void setup(){
  Serial.begin(9600);
  if(Phpoc.begin() == 0){
    Serial.println("Failed to initialize Network");
    for(;;)
      ;
  }
  Phpoc.beginIP6();
  Serial.println(Phpoc.localIP6());
}

void loop(){
}
```

globalIP6()

Description

This function is used to obtains a global IPv6 address of the PHPoC shield for Arduino.

Syntax

Phpoc.globalIP6()

Parameters

none

Returns

Returns a string represents a global IPv6 address. (e.g. 2001:db8:131f::140b)

Example

```
#include <SPI.h>
#include <Phpoc.h>

void setup(){
  Serial.begin(9600);
  if(Phpoc.begin() == 0){
    Serial.println("Failed to initialize Network");
    for(;;)
      ;
  }
  Phpoc.beginIP6();
  Serial.println(Phpoc.globalIP6());
}

void loop(){
}
```

PhpocServer()

Description

Creates a server that listens for incoming connections on the specified port.

Syntax

PhpocServer(port)

Parameters

port - the port to listen on

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.begin();

  Serial.print("Chat server address : ");
  Serial.println(Phpoc.localIP());
}
void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clear out the transmission buffer:
      client.flush();
      Serial.println("We have a new client");
      client.println("Hello, client!");
      alreadyConnected = true;
    }
    if (client.available() > 0) {
      // read the bytes incoming from the client:
      char thisChar = client.read();
      // echo the bytes back to the client:
      server.write(thisChar);
      // echo the bytes to the server as well:
      Serial.write(thisChar);
    }
  }
}
```

begin()

Description

Tells the server to begin listening for incoming connections.

Syntax

server.begin()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.begin();

  Serial.print("Chat server address : ");
  Serial.println(Phpoc.localIP());
}
void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clear out the transmission buffer:
      client.flush();
      Serial.println("We have a new client");
```

```
        client.println("Hello, client!");
        alreadyConnected = true;
      }
      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

beginTelnet()

Description

Tells the server to begin listening for incoming TELNET connections.

Syntax

server.beginTelnet()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  // beginTelnet() enables telnet option negotiation & "character at a time".
  // In "character at a time" mode, text typed is immediately sent to server.
  server.beginTelnet();

  Serial.print("Telnet server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
        // clear out the transmission buffer:
        client.flush();
        Serial.println("We have a new client");
        client.println("Hello, client!");
        alreadyConnected = true;
      }

      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

beginWebSocket()

Description

Tells the server to begin listening for an incoming Web Socket connection.

Syntax

server.beginWebSocket()
server.beginWebSocket(path)

Parameters

path - URI of the web socket

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(80);

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.beginWebSocket("remote_push");

  Serial.print("WebSocket server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  if (client) {
    if (client.available() > 0) {
      // read the bytes incoming from the client:
      char thisChar = client.read();
      if(thisChar == 'A')
        Serial.println("button A press");
      if(thisChar == 'a')
        Serial.println("button A release");
      if(thisChar == 'B')
        Serial.println("button B press");
      if(thisChar == 'b')
        Serial.println("button B release");
      if(thisChar == 'C')
        Serial.println("button C press");
      if(thisChar == 'c')
        Serial.println("button C release");
    }
  }
}
```

beginSSL()

Description

Tells the server to begin listening for an incoming SSL connection.

Syntax

server.beginSSL()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(443);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.beginSSL();

  Serial.print("SSL server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clear out the transmission buffer:
      client.flush();
```

```
        Serial.println("We have a new client");
        client.println("Hello, client!");
        alreadyConnected = true;
      }

      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

beginSSH()

Description

Tells the server to begin listening for an incoming SSH connection.

Syntax

server.beginSSH()
server.beginSSH(username, password)

Parameters

username - user's name from the SSH client
password - password from the SSH client

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(22);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.beginSSH("root", "1234");
  //server.beginSSH("", "");
  //server.beginSSH();

  Serial.print("SSH server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
    if (client) {
      if (!alreadyConnected) {
        // clear out the transmission buffer:
        client.flush();
        Serial.println("We have a new client");
        client.println("Hello, client!");
        alreadyConnected = true;
      }

      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

available()

Description

Gets a client that is connected to the server and has data available for reading.

Syntax

server.available()

Parameters

none

Returns

a client object - on success
false - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.begin();

  Serial.print("Chat server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clear out the transmission buffer:
      client.flush();
      Serial.println("We have a new client");
      client.println("Hello, client!");
      alreadyConnected = true;
    }

    if (client.available() > 0) {
      // read the bytes incoming from the client:
      char thisChar = client.read();
      // echo the bytes back to the client:
      server.write(thisChar);
      // echo the bytes to the server as well:
      Serial.write(thisChar);
    }
  }
}
```

write()

Description

Writes data to all the clients connected to a server.
This data is sent as a byte or series of bytes.

Syntax

server.write(val)
server.write(buf, len)

Parameters

val - a value to send as a single byte (byte of char)
buf - an array to send as a series of bytes (byte or char)
len - the length of the buffer

Returns

Returns an int represents the number of bytes written.

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.begin();

  Serial.print("Chat server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
    if (client) {
      if (!alreadyConnected) {
        // clear out the transmission buffer:
        client.flush();
        Serial.println("We have a new client");
        client.println("Hello, client!");
        alreadyConnected = true;
      }

      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

PhpocClient()

Description

Creates a client which can connect to a server with specified internet IP address and port.

Syntax

PhpocClient()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.println("GET / HTTP/1.0");
    client.println();
  }
  else
    Serial.println("connection failed");
}

void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

connected()

Description

Checks if the client is connected or not.
Note that a client might be considered connected if there is still unread data although the connection has been closed.

Syntax

client.connected()

Parameters

none

Returns

true - when client is connected
false - when client is not connected

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.println("GET / HTTP/1.0");
    client.println();
  }
  else
    Serial.println("connection failed");
}
```

```
void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

connect()

Description

Connects to a server with specified IP address(or hostname) and port.

Syntax

client.connect(ip_addr, port)
client.connect(hostname, port)

Parameters

ip_addr - the IP address which the client will connect to
port - the port number that the client will connect to
hostname - the hostname that the client will connect to

Returns

1 - on success
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.println("GET / HTTP/1.0");
    client.println();
  }
  else
    Serial.println("connection failed");
}
```

```
void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

connectSSL()

Description

Connects to an SSL server with specified IP address(or hostname) and port.

Syntax

client.connectSSL(ip_addr, port)
client.connectSSL(hostname, port)

Parameters

ip_addr - the IP address which the SSL client will connect to
port - the port number that the SSL client will connect to
hostname - the hostname that the SSL client will connect to

Returns

1 - on success
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server[] = "www.arduino.cc";

PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
  ;

  Serial.println("PHPoC SSL Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connectSSL(server, 443)) {
    Serial.println("Connected to server");
    // Make a HTTP request:
    client.println("GET /asciilogo.txt HTTP/1.1");
    client.println("Host: www.arduino.cc");
    client.println("Connection: close");
    client.println();
    Serial.println("Request sent");
  }
}

void loop() {
  // if there are incoming bytes available
  // from the server, read them and print them:
  while (client.available()) {
    char c = client.read();
    Serial.write(c);
  }

  // if the server's disconnected, stop the client:
  if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting from server.");
    client.stop();

    // do nothing forevermore:
    while (true);
  }
}
```

write()

Description

Writes data to the server the client is connected to.
This data is sent as a byte or series of bytes.

Syntax

client.write(val)
client.write(buf, len)

Parameters

val - a value to send as a single byte (byte of char)
buf - an array to send as a series of bytes (byte or char)
len - the length of the buffer

Returns

Returns an int represents the number of bytes written.

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.write("GET / HTTP/1.0₩r₩n", 16);
    client.println();
  }
  else
    Serial.println("connection failed");
}
```

```
void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

available()

Description

Returns the number of bytes available to read from the server which is connected to.

Syntax

client.available()

Parameters

none

Returns

Returns an int represents the number of bytes available.

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.write("GET / HTTP/1.0₩r₩n", 16);
    client.println();
  }
  else
    Serial.println("connection failed");
}

void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

read()

Description

Reads the next byte received from the server the client is connected to.

Syntax

client.read()

Parameters

none

Returns

the next byte - on success
-1 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.write("GET / HTTP/1.0₩r₩n", 16);
    client.println();
  }
  else
    Serial.println("connection failed");
}

void loop() {
  if(client.available())
```

```
    {
      char c = client.read();
      Serial.print(c);
    }

    if(!client.connected())
    {
      Serial.println("disconnected");
      client.stop();
      while(1)
        ;
    }
  }
```

readLine()

Description

Reads line based data from the server the client is connected to.
The line based data means data are finished to CR(0x0d) and LF(0x0a).

Syntax

client.readLine()
client.readLine(buf, size)

Parameters

buf - buffer to store reading data
size - length(bytes) of buffer

Returns

the length of line - on success
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(80);

char slideName;
int slideValue;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.beginWebSocket("remote_slide");

  Serial.print("WebSocket server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();
```

```
    if (client) {
      String slideStr = client.readLine();

      if(slideStr)
      {
        slideName = slideStr.charAt(0);
        slideValue = slideStr.substring(1).toInt();

        Serial.print(slideName);
        Serial.print('/');
        Serial.println(slideValue);
      }
    }
  }
```

flush()

Description

Waits until all outgoing data in buffer have been sent.

Syntax

client.flush()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocServer server(23);
boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  server.begin();

  Serial.print("Chat server address : ");
  Serial.println(Phpoc.localIP());
}

void loop() {
  // wait for a new client:
  PhpocClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clear out the transmission buffer:
      client.flush();
```

```
        Serial.println("We have a new client");
        client.println("Hello, client!");
        alreadyConnected = true;
      }

      if (client.available() > 0) {
        // read the bytes incoming from the client:
        char thisChar = client.read();
        // echo the bytes back to the client:
        server.write(thisChar);
        // echo the bytes to the server as well:
        Serial.write(thisChar);
      }
    }
  }
```

stop()

Description

Disconnects from the server.

Syntax

client.stop()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

char server_name[] = "www.arduino.cc";
PhpocClient client;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Serial.println("PHPoC TCP Client test");

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET);
  //Phpoc.begin();

  if(client.connect(server_name, 80))
  {
    Serial.println("connected");
    client.println("GET / HTTP/1.0");
    client.println();
  }
  else
    Serial.println("connection failed");
}

void loop() {
  if(client.available())
  {
    char c = client.read();
    Serial.print(c);
  }

  if(!client.connected())
  {
    Serial.println("disconnected");
    client.stop();
    while(1)
      ;
  }
}
```

setOutgoingServer()

Description

Sets the outgoing mail server.

Syntax

email.setOutgoingServer(hostname, port)

Parameters

hostname - hostname of the outgoing mail server
port - port number of the outgoing mail server

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test using outgoing relay server");

  // [login using your private password]
  // Google may block sign-in attempts from some apps or devices that do not use modern security
standards.
  // Change your settings to allow less secure apps to access your account.
  // https://www.google.com/settings/security/lesssecureapps

  // [login using app password]
  // 1. turn on 2-step verification
  // 2. create app password
  // 3. apply app password as your login password

  // setup outgoing relay server - gmail.com
  email.setOutgoingServer("smtp.gmail.com", 587);
  email.setOutgoingLogin("your_login_id", "your_login_password or app_password");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

setOutgoingLogin()

Description

Sets log in information of the outgoing mail server.

Syntax

email.setOutgoingLogin(id, password)

Parameters

id - username or id of the account
password - password of the account

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test using outgoing relay server");

  // [login using your private password]
  // Google may block sign-in attempts from some apps or devices that do not use modern security
standards.
  // Change your settings to allow less secure apps to access your account.
  // https://www.google.com/settings/security/lesssecureapps

  // [login using app password]
  // 1. turn on 2-step verification
  // 2. create app password
  // 3. apply app password as your login password

  // setup outgoing relay server - gmail.com
  email.setOutgoingServer("smtp.gmail.com", 587);
  email.setOutgoingLogin("your_login_id", "your_login_password or app_password");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

setFrom()

Description

Sets a sender's e-mail address and name

Syntax

email.setFrom(email_addr, name)

Parameters

email_addr - sender's e-mail address
name - sender's name

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

setTo()

Description

Sets a receiver's e-mail address and name

Syntax

email.setTo(email_addr, name)

Parameters

email_addr - receiver's e-mail address
name - receiver's name

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

setSubject()

Description

Sets a subject of the e-mail.

Syntax

email.setSubject(subject)

Parameters

subject - subject of the e-mail

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
```

```
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

beginMessage()

Description

Gets ready to put contents into the e-mail body.
The write(), print() or println() can be used for writing the e-mail after calling this function.
To end writing e-mail, endMassage() is used.

Syntax

email.beginMessage()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();
  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

endMessage()

Description

Gets finished putting contents into the e-mail body.

> Calling this function after writing messages is highly recommended. If you don't use this function, you may loose the last line of the messages.

Syntax

email.endMessage()

Parameters

none

Returns

none

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.println("Hello, world!");
  email.println("I am PHPoC Shield for Arduino");
  email.println("Good bye");
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

write()

Description

Writes data to e-mail body.
This data can be written as a byte or series of bytes.

Syntax

email.write(val)
email.write(buf, len)

Parameters

val - a value to write as a single byte
buf - an array to write as series of byes
len - the length of the buffer

Returns

Returns an int represents the number of bytes written.

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.write("H");
  email.write("ello₩r₩n", 6);
  email.endMessage();
  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

send()

Description

Sends an e-mail.

> Before calling this function, you need to set required parameters such as e-mail addresses of receiver and sender.

Syntax

email.send()

Parameters

none

Returns

1 - on success
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocEmail email;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin(PF_LOG_SPI | PF_LOG_NET | PF_LOG_APP);
  //Phpoc.begin();

  Serial.println("Email Client Test");

  // setup From/To/Subject
  email.setFrom("from_email_address", "from_user_name");
  email.setTo("to_email_address", "to_user_name");
  email.setSubject("Mail from PHPoC Shield for Arduino");

  // write email message
  email.beginMessage();
  email.write("H");
  email.write("ello₩r₩n", 6);
  email.endMessage();

  // send email
  if(email.send() > 0)
    Serial.println("Email send ok");
  else
    Serial.println("Email send failed");
}

void loop() {
}
```

date()

Description

Gets the current date and time from RTC of PHPoC Shield for Arduino.

Syntax

datetime.date()
datetime.date(format)

Parameters

format - time format

Format	Description
Y	A full numeric representation of a year, 4 digits (example: 2016)
y	A two digit representation of a year (example: 16)
M	A short texture representation of a month, three letters (example: Mar)
m	Numeric representation of a month with leading zeros (example: 03)
n	Numeric representation of a month without leading zeros (example: 3)
d	Day of the month, 2 digits with leading zeros (01 to 31)
j	Day of the month without leading zeros (1 to 31)
D	A textual representation of a day, three letters (example: Mon)
g	12-hour format of an hour without leading zeros (1 to 12)
G	24-hour format of an hour without leading zeros (0 to 23)
h	12-hour format of an hour with leading zeros (01 to 12)
H	24-hour format of an hour with leading zeros (00 to 23)
i	Minutes with leading zeros (00 to 59)
s	Seconds with leading zeros (00 to 59)
a	Lowercase Ante meridiem and Post meridiem (am or pm)
A	Uppercase Ante meridiem and Post meridiem (AM or PM)

Returns

With a given format, it returns a formatted string represents the current date and time.
Without a given format, it returns a string with the last given format.
The default format is "D M j H:i:s".

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
```

```
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();

  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

year()

Description

Gets the current year from RTC of PHPoC Shield for Arduino.

Syntax

datetime.year()

Parameters

none

Returns

the current year - on success (2000 ~ 2099)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

month()

Description

Gets the current month from RTC of PHPoC Shield for Arduino.

Syntax

datetime.month()

Parameters

none

Returns

the current month - on success (1~12)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

day()

Description

Gets the current day from RTC of PHPoC Shield for Arduino.

Syntax

datetime.day()

Parameters

none

Returns

the current day - on success (1~31)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

dayofWeek()

Description

Gets the current day of week from RTC of PHPoC Shield for Arduino.

Syntax

datetime.dayofWeek()

Parameters

none

Returns

the current day of week - on success (1 ~ 7)

Return	Day of Week
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();

  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

hour()

Description

Gets the current hour from RTC of PHPoC Shield for Arduino.

Syntax

datetime.hour()

Parameters

none

Returns

the current hour - on success (0 ~ 23)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

minute()

Description

Gets the current minute from RTC of PHPoC Shield for Arduino.

Syntax

datetime.minute()

Parameters

none

Returns

the current minute - on success (0 ~ 59)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```

second()

Description

Gets the current second from RTC of PHPoC Shield for Arduino.

Syntax

datetime.second()

Parameters

none

Returns

the current second - on success (0 ~ 59)
0 - on failure

Example

```
#include <SPI.h>
#include <Phpoc.h>

PhpocDateTime datetime;

void setup() {
  Serial.begin(9600);
  while(!Serial)
    ;

  Phpoc.begin();

  Serial.println("Phpoc Time test");

  Serial.print(datetime.year());
  Serial.print('-');
  Serial.print(datetime.month());
  Serial.print('-');
  Serial.print(datetime.day());
  Serial.print(' ');
  Serial.print(datetime.dayofWeek());
  Serial.print(':');
  Serial.print(datetime.hour());
  Serial.print(':');
  Serial.print(datetime.minute());
  Serial.print(':');
  Serial.print(datetime.second());
  Serial.println();
  datetime.date("Y-m-d H:i:s");
}

void loop() {
  Serial.println(datetime.date());
  delay(1000);
}
```