ShieldInterface Class

Overview

The ShieldInterface class provides low-level functions for sending data and commands to the Telstra IoT Shield and receiving responses. This class is used extensively by higher-level classes such as IoTShield and Connection4G. Its member functions are not intended to be called directly from Arduino sketches.

Constructor

ShieldInterface()

Parameters
None.

Return Value
None.

Description
Creates a ShieldInterface instance, initialises the SPI bus and sets up the chip select and data ready pins for communication with the IoT shield.

Public Member Functions

int sendCommand(char *buffer, int length)

Parameters

char *buffer	Buffer containing command packet [max 2048 bytes]
int length	Length of command packet in buffer

Return Value
Number of bytes sent.

Description
Sends a packet to the IoT shield and reports number of bytes written.

int getResponse(char *buffer, int length, int timeout)

Parameters

char *buffer	Buffer to receive response packet [min 2048 bytes]
int length	Length of response buffer
int timeout	Number of milliseconds to wait for response

Return Value
Number of bytes received (0 if timeout occurred).

Description
Waits for a response from the IoT shield, then saves response to buffer and reports number of bytes received. If no response is received within timeout period, returns zero (buffer will be unmodified).

Connection4G Class

Overview

The Connection4G class handles internet connectivity over a 4G network using the IoT Shield. This class is used by the TelstraIoT class to connect to the Telstra IoT platform, but its member functions can also be called directly from Arduino sketches.

Constructor

Connection4G(bool secure, ShieldInterface *shield)

Parameters

bool secure	Boolean value indicating whether TLS will be used
ShieldInterface *shield	Instance of ShieldInterface used to communicate with IoT shield

Return Value

None.

Description

Creates a Connection4G instance. If 'secure' is set to true, TLS will be used to secure all TCP connections made using Connection4G. The Telstra IoT Platform ONLY accepts secure connections so 'secure' must be set to true if this class is to be used with the TelstraIoT class.

Public Member Functions

int ping (const char *host, char *responseBuffer)

Parameters

const char *host	Host name or IP address [max 256 bytes]
char *responseBuffer	Buffer to hold host IP address resolved by IoT shield [min 16 bytes]

Return Value

Response time of host in milliseconds. Returns CONNECTION4G_STATUS_ERROR (-1) either the host or the IoT shield did not respond.

Description

Pings the specified host and reports the response time. Also saves the host's resolved IP address to a buffer. If either the host or the shield does not respond within 5 seconds, the function will return CONNECTION4G_STATUS_ERROR (-1) and the buffer contents can be considered invalid.

int openTCP(const char *host, uint16_t port)

Parameters

const char *host	Host name or IP address [max 256 bytes]
uint16_t port	Port for TCP connection (typically 80 for HTTP, 443 for HTTPS)

Return Value

Integer status code:
CONNECTION4G_STATUS_OK (0) if connection was successfully established
CONNECTION4G_STATUS_ERROR (-1) if an error occurred or shield was not ready.

Description

Opens a TCP connection to the specified host on the specified port. The connection will be established using TLS if 'secure' was set to true when Connection4G was initialised. Only one TCP connection can be open through the IoT shield at any time.

int closeTCP()

Parameters

None.

Return Value

Integer status code:
CONNECTION4G_STATUS_OK (0) if connection was successfully closed
CONNECTION4G_STATUS_ERROR (-1) if the shield did not respond within 15 seconds.

Description

Closes a TCP connection previously opened with openTCP().

int TCPRead(char *buffer, uint16_t length)

Parameters

char *buffer	Buffer to receive data [min 2048 bytes]
uint16_t length	Length of data buffer

Return Value

Returns:
Length of data received (in bytes), if command was successful.
CONNECTION4G_STATUS_ERROR (-1) if no data was received or a timeout occurred.

Description

Listens for incoming data on a TCP connection and saves it to a buffer. If no data is received in 5 seconds or an error occurs, the function will return an error code and the data in the buffer should be considered invalid.

int TCPWrite(char *data, uint16_t length)

Parameters

char *data	Buffer of data to send [max 2048 bytes]
uint16_t length	Length of data buffer

Return Value

Integer status code:
CONNECTION4G_STATUS_OK (0) if data was successfully sent
CONNECTION4G_STATUS_ERROR (-1) if the shield did not respond within 5 seconds.

Description

Sends data to host over a TCP connection.

int getSignalQuality(char *data)

Parameters

char *data	Buffer to save result [min 8 bytes]

Return Value

Integer status code:
CONNECTION4G_STATUS_OK (0) if report was received successfully
CONNECTION4G_STATUS_ERROR (-1) if an error or timeout occurred.

Description

Gets a signal quality report from the 4G modem on the IoT shield. A string will be saved to the data buffer provided containing the current signal level in dBm (this will typically be in the range -120dBm to -50dBm). If no data is received in 5 seconds or an error occurs, the function will return an error code and the data in the buffer should be considered invalid.

int activatePDP(const char *apn, const char *username, const char *password)

Parameters

char *apn	APN to use for PDP context [max 128 bytes]
char *username	Username to use for PDP context [max 128 bytes]
char *password	Password to use for PDP context [max 128 bytes]

Return Value

Integer status code:
CONNECTION4G_STATUS_OK (0) if command was successfully sent to modem
CONNECTION4G_STATUS_ERROR (-1) if an error or timeout occurred.

Description

Activates a new PDP context using the specified APN. Username and password should be empty strings unless login details are required for specified APN.

NOTE: Manually establishing a PDP context using this function is not required for standard use. When powered on, the shield will automatically establish a PDP context using Telstra's default APN.

int deactivatePDP()

Parameters
None.

Return Value
Integer status code:
CONNECTION4G_STATUS_OK (0) if command was successfully sent to modem
CONNECTION4G_STATUS_ERROR (-1) if an error or timeout occurred.

Description
Deactivates a PDP previously opened with activatePDP().

IoTShield Class

Overview

The IoTShield class handles communication with the IoT shield and its onboard sensors. This class is used by the TelstraIoT class to generate accurate timestamps for communication with the Telstra IoT Platform, but its member functions can also be called directly from Arduino sketches.

Constructor

IoTShield(ShieldInterface *shield)

Parameters

ShieldInterface *shield Instance of ShieldInterface used to communicate with IoT shield

Return Value

None.

Description

Creates an IoTShield instance.

Public Member Functions

void getIMEI(char* resultBuffer)

Parameters

char *resultBuffer Buffer to hold IMEI [min 16 bytes]

Return Value

None.

Description

Requests the IMEI (International Mobile Equipment Identity) of the IoT Shield and saves the result to a buffer.

void getIP(char* resultBuffer)

Parameters

char *resultBuffer Buffer to hold IP address [min 16 bytes]

Return Value

None.

Description

Requests the public IP address of the IoT Shield and saves the result to a buffer.

void getTime(char* resultBuffer)

Parameters

char *resultBuffer Buffer to hold time string [min 50 bytes]

Return Value

None.

Description

Requests the time and date from the real time clock (RTC) on the IoT Shield and saves the result to a buffer.

bool isEC21Connected()

Parameters

None.

Return Value

Boolean value indicating 4G modem status.

Description

Returns true if the microcontroller on the shield is able to communicate with the 4G modem, false otherwise.

bool isSIMReady()

Parameters

None.

Return Value

Boolean value indicating SIM status.

Description

Returns true if SIM card is inserted and valid, otherwise returns false.

bool isPDPContextActive()

Parameters

None.

Return Value

Boolean value indicating PDP context status.

Description

Returns true if PDP context is active, otherwise returns false.

bool isShieldReady()

Parameters
None.

Return Value
Boolean value indicating shield status.

Description
Returns true if shield is ready to accept commands, otherwise returns false.

void waitUntilShieldIsReady()

Parameters
None.

Return Value
None.

Description
Waits until shield is ready to receive commands and has an active, valid PDP context.

void writeCredentials(const char *deviceID, const char *deviceTenant, const char *deviceUsername, const char *devicePassword)

Parameters

const char *deviceID	Device ID string (generated when device is registered on the Telstra IoT Platform). [max 8 bytes]
const char *deviceTenant	Tenant ID for the Telstra IoT Platform. [max 32 bytes]
const char *deviceUsername	Username for the Telstra IoT Platform. [max 32 bytes]
const char *devicePassword	Password for the Telstra IoT Platform. [max 32 bytes]

Return Value
None.

Description
Writes credentials for the Telstra IoT platform to secure, non-volatile memory on the IoT shield. Credentials are saved automatically (using this function) by the registerDevice() function in the TelstraIoT class, so calling it manually is generally not necessary.

void readCredentials(char *deviceID, char *deviceTenant, char *deviceUsername, char *devicePassword)

Parameters

char *deviceID	Buffer to hold device ID string [min 8 bytes]
char *deviceTenant	Buffer to hold tenant ID [min 32 bytes]
char *deviceUsername	Buffer to hold username [min 32 bytes]
char *devicePassword	Buffer to hold password [min 32 bytes]

Return Value
None.

Description
Reads saved credentials from the IoT shield into the buffers provided.

void clearCredentials()

Parameters
None.

Return Value
None.

Description
Deletes saved credentials from the IoT shield.

void resetModem()

Parameters
None.

Return Value
None.

Description
Resets the 4G modem on the IoT shield. Modem will reboot and reconnect to network within 30 seconds.

int getBatteryStatus()

Parameters
None.

Return Value
Integer code representing battery status.
1 -Disconnected
2 - Connected, powering system
3 - Connected, charging

Description
Gets battery status information from the IoT shield.

int getBatteryStateOfCharge()

Parameters
None.

Return Value
Battery state of charge as percentage (0-100)

Description
Gets battery state of charge information from the IoT shield.

void getTemperature(char *temperature)

Parameters
char *temperature Buffer to hold temperature measurement [min 6 bytes]

Return Value
None.

Description
Gets temperature measurement from onboard sensor on IoT shield. Result will be a string containing the temperature in degrees Celsius to two decimal places.

void getLightLevel(char *lightLevel)

Parameters
char *lightLevel Buffer to hold light measurement [min 8 bytes]

Return Value
None.

Description
Gets light level measurement from onboard sensor on IoT shield. Result will be a string containing the measured light level in lux.

void updateRTCFromNetwork(char *resultBuffer)

Parameters
char *resultBuffer Buffer to hold time [min 50 bytes]

Return Value
None.

Description
Updates the onboard real-time clock (RTC) with time from the 4G network, and saves the current time to the buffer provided. This command is automatically run each time the shield is powered on.

TelstraIoT Class

Overview

The TelstraIoT class handles communication with the Telstra IoT Platform. This high-level class makes use of both the Connection4G and IoTShield classes.

Constructors

TelstraIoT(Connection4G* conn, IoTShield* shield)

Parameters

Connection4G* conn	Instance of Connection4G to use for network connectivity
IoTShield* shield	Instance of ShieldInterface used to communicate with IoT shield

Return Value

None.

Description

Creates a TelstraIoT instance WITHOUT setting up credentials or hostname. If this constructor is used, the SetCredentials and SetHost functions MUST be called before attempting to connect to the Telstra IoT Platform. This constructor variant is intended for use on a shield which has already been registered on the Telstra IoT Platform and has credentials saved to the IoT Shield.

TelstraIoT(const char* host, const int port, const char* tenantId, const char* user, const char* password, const char* applicationKey, Connection4G* conn, IoTShield* shield)

Parameters

const char* host	Hostname for Telstra IoT Platform [max 256 bytes]
const int port	Port for connection (typically 443)
const char* tenantId	Tenant ID for Telstra IoT Platform [max 32 bytes]
const char* user	Username for Telstra IoT Platform [max 32 bytes]
const char* password	Password for Telstra IoT Platform [max 32 bytes]
const char* applicationKey	Application Key for Telstra IoT Platform [max 32 bytes]
Connection4G* conn	Instance of Connection4G to use for network connectivity
IoTShield* shield	Instance of ShieldInterface used to communicate with IoT shield

Return Value

None.

Description

Creates a TelstraIoT instance and sets up hostname, credentials, etc. This constructor variant is best used when registering a new device on the Telstra IoT Platform. Application key can be empty string (“”) if not needed.

TelstraIoT(const char* host, const char* tenantId, const char* user, const char* password, const char* applicationKey, Connection4G* conn, IoTShield* shield)

Parameters

const char* host	Hostname for Telstra IoT Platform [max 256 bytes]
const char* tenantId	Tenant ID for Telstra IoT Platform [max 32 bytes]
const char* user	Username for Telstra IoT Platform [max 32 bytes]
const char* password	Password for Telstra IoT Platform [max 32 bytes]
const char* applicationKey	Application Key for Telstra IoT Platform [max 32 bytes]
Connection4G* conn	Instance of Connection4G to use for network connectivity
IoTShield* shield	Instance of ShieldInterface used to communicate with IoT shield

Return Value

None.

Description

Creates a TelstraIoT instance and sets up hostname, credentials, etc. Uses default port for HTTPS (port 443). This constructor variant is best used when registering a new device on the Telstra IoT Platform.

Public Member Functions

int raiseAlarm(const char* type, const char* status, const char* severity, const char* time, const char* text)

Parameters

const char* type	Alarm type [max 256 bytes]
const char* status	Alarm status (ACTIVE/ACKNOWLEDGED/CLEARED) [max 32 bytes]
const char* severity	Alarm severity (WARNING/MINOR/MAJOR/CRITICAL) [max 32 bytes]
const char* time	Time of alarm event [max 256 bytes]
const char* text	Text description of alarm event [max 256 bytes]

Return Value

Integer status code:
1 - Alarm message was sent successfully
-2 - Alarm message was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends an alarm notification to the Telstra IoT Platform.

int raiseAlarm(const char* type, const char* status, const char* severity, const char* text)

Parameters

const char* type	Alarm type [max 256 bytes]
const char* status	Alarm status (ACTIVE/ACKNOWLEDGED/CLEARED) [max 32 bytes]
const char* severity	Alarm severity (WARNING/MINOR/MAJOR/CRITICAL) [max 32 bytes]
const char* text	Text description of alarm event [max 256 bytes]

Return Value

Integer status code:
1 - Alarm message was sent successfully
-2 - Alarm message was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends an alarm notification to the Telstra IoT Platform. The timestamp of the alarm notification will be automatically obtained from the RTC on the IoT shield.

int sendMeasurement(const char* type, const char* time, const char* fragmentName, const char* measurementName, const long &mValue, const char* mUnit)

Parameters

const char* type	Measurement type [max 256 bytes]
const char* time	Measurement timestamp [max 256 bytes]
const char* fragmentName	Measurement fragment name [max 256 bytes]
const char* measurementName	Measurement name [max 256 bytes]
const long &mValue	Pointer to long integer containing measurement value
const char* mUnit	Measurement unit [max 256 bytes]

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends an integer measurement to the Telstra IoT Platform.

int sendMeasurement(const char* type, const char* fragmentName, const char* measurementName, const long &mValue, const char* mUnit)

Parameters

const char* type	Measurement type [max 256 bytes]
const char* fragmentName	Measurement fragment name [max 256 bytes]
const char* measurementName	Measurement name [max 256 bytes]
const long &mValue	Pointer to long integer containing measurement value
const char* mUnit	Measurement unit [max 256 bytes]

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends an integer measurement to the Telstra IoT Platform. Timestamp will be automatically obtained from the RTC on the IoT shield.

int sendMeasurement(const char* type, const char* time, const char* fragmentName, const char* measurementName, const long &mValue, const char* mUnit)

Parameters

const char* type	Measurement type [max 256 bytes]
const char* time	Measurement timestamp [max 256 bytes]
const char* fragmentName	Measurement fragment name [max 256 bytes]
const char* measurementName	Measurement name [max 256 bytes]
char* measurementString	String containing measurement [max 256 bytes]
const char* mUnit	Measurement unit [max 256 bytes]

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends a measurement (stored as a string) to the Telstra IoT Platform.

int sendMeasurement(const char* type, const char* fragmentName, const char* measurementName, const long &mValue, const char* mUnit)

Parameters

const char* type	Measurement type [max 256 bytes]
const char* fragmentName	Measurement fragment name [max 256 bytes]
const char* measurementName	Measurement name [max 256 bytes]
char* measurementString	String containing measurement [max 256 bytes]
const char* mUnit	Measurement unit [max 256 bytes]

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Sends a measurement (stored as a string) to the Telstra IoT Platform. Timestamp will be automatically obtained from the RTC on the IoT shield.

int registerDevice(const char* name, char* id, const int idSize, const char** supportedOperations, const int nSupportedOperations, const char** supportedMeasurements, const int nSupportedMeasurements)

Parameters

const char* name	Device name [max 256 bytes]
char* id	Buffer to save device ID [min 8 bytes]
const int idSize	Size of ID buffer
const char** supportedOperations	Array of strings containing list of supported operations [max 512 bytes]
const int nSupportedOperations	Number of supported operations
const char** supportedMeasurements	Array of strings containing list of supported measurement types [max 512 bytes]
const int nSupportedMeasurements	Number of supported measurement types

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Registers a device on the Telstra IoT Platform and sends lists of supported operations and measurements. NOTE: Operations are currently not supported in the TelstraIoT library.

int registerDevice(const char* name, char* id, const int idSize, const char** supportedMeasurements, const int nSupportedMeasurements)

Parameters

const char* name	Device name [max 256 bytes]
char* id	Buffer to save device ID [min 8 bytes]
const int idSize	Size of ID buffer
const char** supportedMeasurements	Array of strings containing list of supported measurement types [max 512 bytes]
const int nSupportedMeasurements	Number of supported measurement types

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Registers a device on the Telstra IoT Platform and sends list of supported measurements.

int registerDevice(const char* name, char* id, const int idSize)

Parameters

const char* name	Device name [max 256 bytes]
char* id	Buffer to save device ID [min 8 bytes]
const int idSize	Size of ID buffer

Return Value

Integer status code:
1 - Measurement was sent successfully
-2 - Measurement was sent but no response was received from the server
Other negative values - HTTP status code of server's response multiplied by negative 1

Description

Registers a device on the Telstra IoT Platform with no supported operations or measurement types.

void setCredentials(const char* _deviceId, const char* _tenantId, const char* _user, const char* _password, const char* _applicationKey)

Parameters

const char* _deviceId	Device ID string [max 8 bytes]
const char* _tenantId	Tenant ID string [max 32 bytes]
const char* _user	Username for Telstra IoT platform [max 32 bytes]
const char* _password	Password for Telstra IoT platform [max 32 bytes]
const char* _applicationKey	Application key for Telstra IoT platform [max 32 bytes]

Return Value
None.

Description
Sets the credentials associated with this TelstraIoT instance. This is necessary if credentials were not provided when instance was initialised.

void setHost(const char* _host, const int _port)

Parameters

const char* _host	Hostname of Telstra IoT Platform [max 256 bytes]
const int _port	Port for connection (typically 443)

Return Value
None.

Description
Sets the hostname and port associated with this TelstraIoT instance. This is necessary if the hostname was not specified when this instance was initialised.

void writeCredentials(const char *deviceID, const char *deviceTenant, const char *deviceUsername, const char *devicePassword)

Parameters

const char *deviceID	Device ID string (generated when device is registered on the Telstra IoT Platform). [max 8 bytes]
const char *deviceTenant	Tenant ID for the Telstra IoT Platform. [max 32 bytes]
const char *deviceUsername	Username for the Telstra IoT Platform. [max 32 bytes]
const char *devicePassword	Password for the Telstra IoT Platform. [max 32 bytes]

Return Value
None.

Description
Writes credentials for the Telstra IoT platform to secure, non-volatile memory on the IoT shield. Credentials are saved automatically (using this function) by the registerDevice() function in the TelstraIoT class, so calling it manually is generally not necessary.

void readCredentials(char *deviceID, char *deviceTenant, char *deviceUsername, char *devicePassword)

Parameters

char *deviceID	Buffer to hold device ID string [min 8 bytes]
char *deviceTenant	Buffer to hold tenant ID [min 32 bytes]
char *deviceUsername	Buffer to hold username [min 32 bytes]
char *devicePassword	Buffer to hold password [min 32 bytes]

Return Value

None.

Description

Reads saved credentials from the IoT shield into the buffers provided. Can be used in conjunction with SetCredentials to load saved credentials and use them with this TelstraIoT instance.

void clearCredentials()

Parameters

None.

Return Value

None.

Description

Deletes saved credentials from the IoT shield.

END OF DOCUMENT