## Obsługa protokołu MQTT

Do obsługi protokołu przez aplikację jest zbudowana przez nas osobna biblioteka. Umożliwia ona zarówno subskrypcję jak i publikowanie informacji. Dzięki temu biblioteka może być stosowana do programu sterującego czujnikiem oraz aplikacji z której korzysta użytkownik. Inne funkcje tej biblioteki to: nawiązywanie połączenia, aktualizowanie logu, obsługa GUI.

## Kluczowe funkcje biblioteki

Podstawową metodą jest ToggleConnection(). Zależnie od stanu, połączenie jest nawiązywane lub zamykane.

void MqttLib::ToggleConnection()

{

if (m\_client->state() == QMqttClient::Disconnected) {

m\_client->connectToHost();

} else {

m\_client->disconnectFromHost();

}

}

Kolejną funkcjonalnością jest publikowanie informacji za pomocą Publish. Jeśli informacja zostanie poprawnie wysłana, zwraca wartość true. W przeciwnym wypadku zwraca wartość false.

bool MqttLib::Publish(QMqttTopicName &name, QByteArray &message)

{

if (m\_client->publish(name, message) == -1)

return false;

return true;

}

Subskrypcja informacji realizowana jest za pomocą Subscribe. Wyjściem funkcji jest wskaźnik na obiekt subskrybowany.

QMqttSubscription\* MqttLib::Subscribe(QMqttTopicFilter &filter)

{

auto subscription = m\_client->subscribe(filter);

if (!subscription)

return nullptr;

return subscription;

}

## Pozostałe funkcje biblioteki

SetClientPort pozwala na zmianę portu.

void MqttLib::SetClientPort(int p)

{

m\_client->setPort(p);

}

Konstruktor tworzy obiekt QMqttClient do obsługi wszystkiego związanego z przesyłaniem danych za pomocą tego protokołu.

MqttLib::MqttLib()

{

m\_client = new QMqttClient();

}