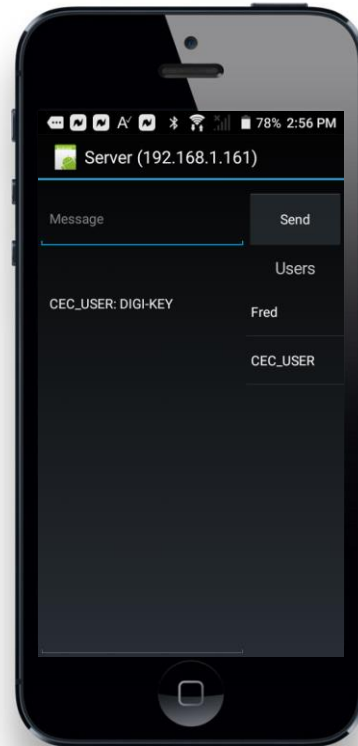


Easy TCP/IP for IoT



MQTT Primer

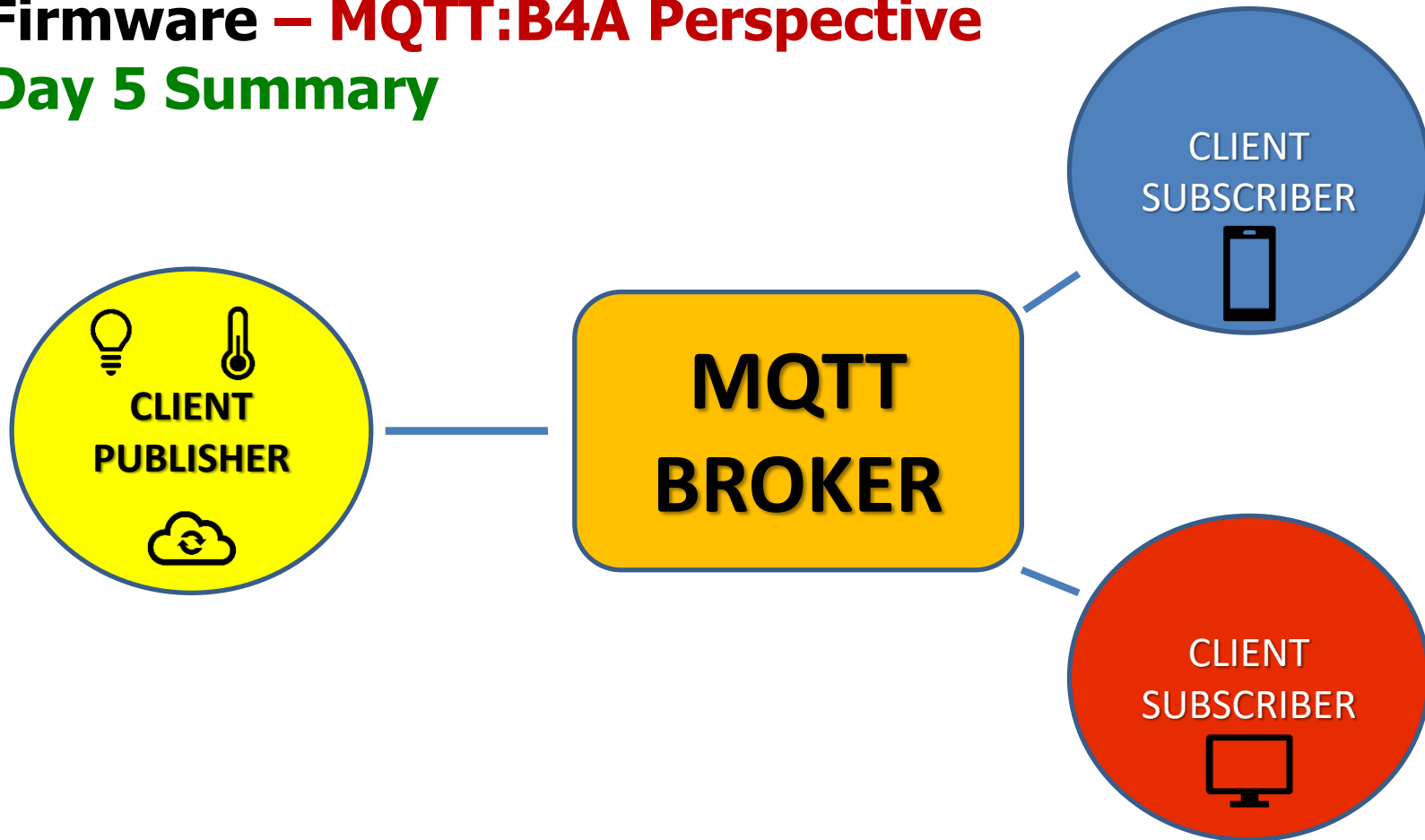
October 25, 2019

Fred Eady

Easy TCP/IP for IoT

AGENDA

- **Firmware – MQTT Basics**
- **Firmware – MQTT:B4A Perspective**
- **Day 5 Summary**

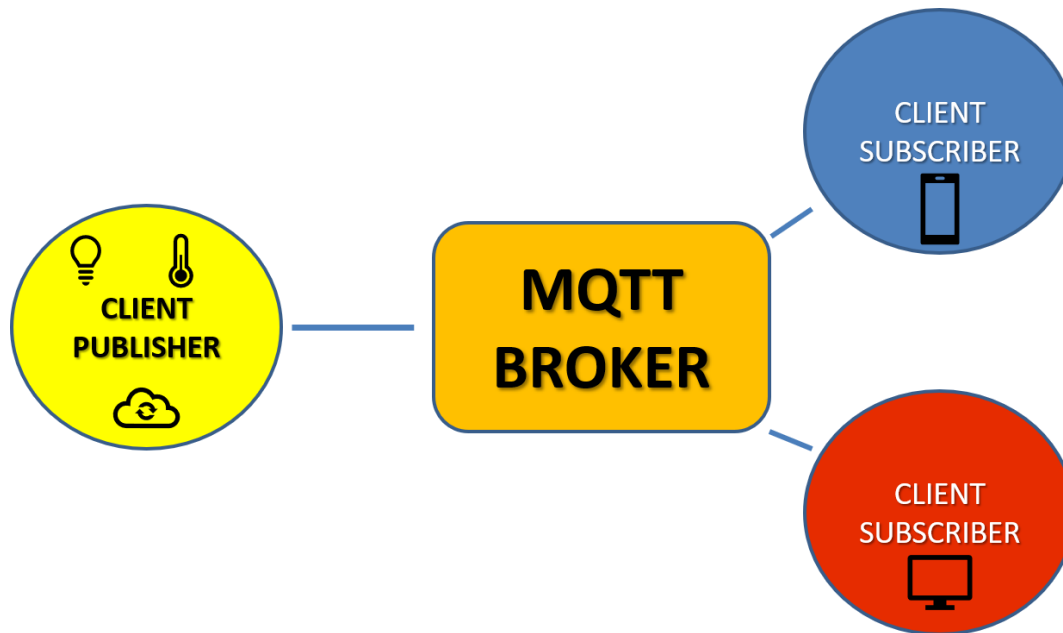


Easy TCP/IP for IoT

Firmware - MQTT Basics

“MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium.”

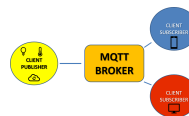
Cited from the official MQTT 3.1.1 specification



Easy TCP/IP for IoT

Firmware – MQTT Basics

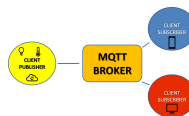
- MQTT was invented in 1999 at IBM.
- MQTT was originally an acronym for MQ Telemetry Transport.
- Clients connect to a Broker.
- The Broker routes messages and manages connections.
- Brokers can be local or online.
- Clients can publish and/or subscribe.
- A client only needs to publish data one time.
- Clients can listen to messages published by other clients.



Easy TCP/IP for IoT

Firmware – MQTT Basics – Publish/Subscribe (Pub/Sub)

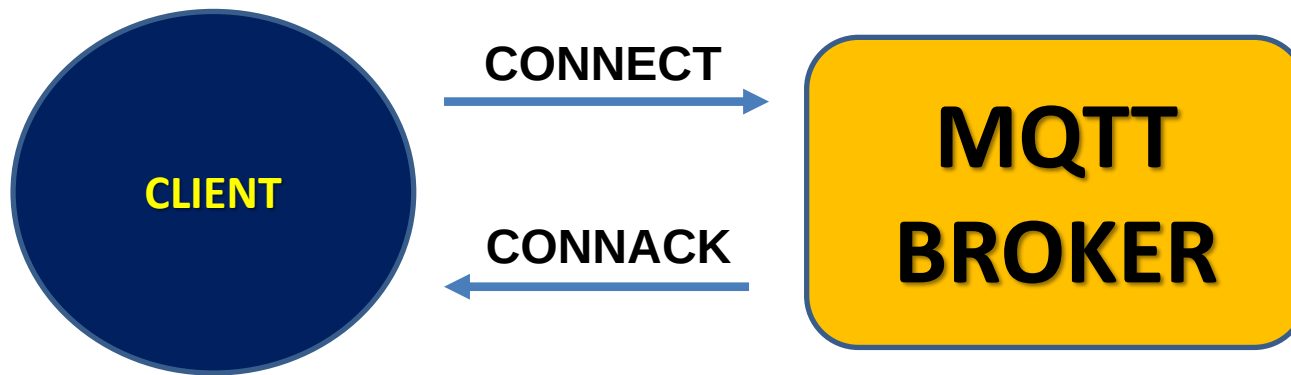
- Pub/Sub is an alternative to the client/server model.
- MQTT publishing clients are decoupled from subscriber clients by a broker.
- MQTT brokers filter messages and distribute them to subscribers.
- Every MQTT message contains a topic (subject).
- MQTT routes messages to subscribers using the topic.



Easy TCP/IP for IoT

Firmware - MQTT Basics - Client/Broker Connections

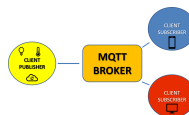
- MQTT connections are between one client and the broker.
- MQTT clients do not connect to each other.
- A client initiates a connection with the **CONNECT** message.
- The broker responds with a **CONNACK** message.



Easy TCP/IP for IoT

Firmware - MQTT Basics - CONNECT Packet

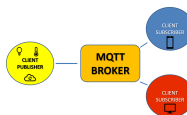
```
clientId      "client_A"  
cleanSession  true  
username      "fred"  
password      "fredpass"  
lastWillTopic "/fred/will"  
lastWillQos   2  
lastWillMessage "unexpected exit"  
lastWillRetain false  
keepAlive     60
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - MQTT Events - Connected

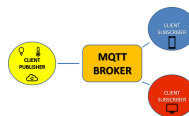
```
2 Sub Process_Globals
3   Private client As MqttClient
4   Private const port As Int = 51042
5   Private serializator As B4XSerializator
6   Public connected As Boolean
7   Private fx As JFX
8   Private currentName As String
9 End Sub
10
11
12
13 Public Sub ConnectTo(Host As String, Name As String)
14   Disconnect
15   currentName = Name
16   client.Initialize("client", $"tcp://{Host}:{port}", "desktop" & Rnd(1, 1000000))
17   Dim mo As MqttConnectOptions
18   mo.Initialize("", "")
19   mo.SetLastWill("all/disconnect", serializator.ConvertObjectToBytes(currentName), 0, False)
20   client.Connect2(mo)
21 End Sub
22
23 Private Sub client_Connected (Success As Boolean)
24   Log($"Connected: ${Success}")
25   If Success Then
26     connected = True
27     client.Subscribe("all/#", 0)
28     client.Publish2("all/connect", serializator.ConvertObjectToBytes(currentName), 0, False)
29   Else
30     fx.Msgbox(Main.MainForm, "Error connecting: " & LastException, "")
31   End If
32 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - MQTT Events - MessageArrived

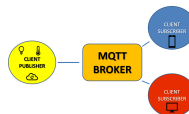
```
34 Private Sub client_MessageArrived (Topic As String, Payload() As Byte)
35     Dim receivedObject As Object = serializer.ConvertBytesToObject(Payload)
36     If Topic = "all/connect" Or Topic = "all/disconnect" Then
37         'new client has connected or disconnected
38         'do nothing. The server will handle it.
39     Else if Topic = "all/users" Then
40         Dim newUsers As List = receivedObject
41         Main.NewUsers(newUsers)
42     Else
43         Dim m As Message = receivedObject
44         Main.NewMessage(m)
45     End If
46
47 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - MQTT Events - Disconnected

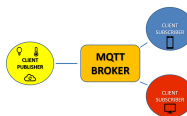
```
63 Public Sub Disconnect
64     If connected Then
65         connected = False
66         client.Publish2("all/disconnect", serializator.ConvertObjectToBytes(currentName), 0, False)
67         client.Close
68     End If
69 End Sub
70
71 Private Sub client_Disconnected
72     If connected Then
73         fx.Msgbox(Main.MainForm, "Disconnected", "")
74     End If
75     connected = False
76 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - Publish

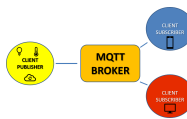
```
49 Private Sub CreateMessage(Body As String) As Byte()  
50     Dim m As Message  
51     m.Initialize  
52     m.Body = Body  
53     m.From = currentName  
54     Return serializer.ConvertObjectToBytes(m)  
55 End Sub  
56  
57 Public Sub SendMessage(Body As String)  
58     If connected Then  
59         client.Publish2("all", CreateMessage(Body), 0, False)  
60     End If  
61 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - Embedded Broker

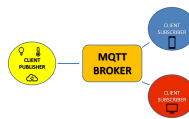
```
6  Sub Process_Globals
7      Private broker As MqttBroker
8      Private client As MqttClient
9      Private const port As Int = 51042
10     Private serializer As B4XSerializer
11     Public connected As Boolean
12     Private brokerStarted As Boolean
13     Private users As List
14     Public isServer As Boolean
15     Private currentName As String
16 End Sub
17
18 Sub Service_Create
19     broker.Initialize("", port)
20     broker.DebugLog = False
21     users.Initialize
22 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - Embedded Broker

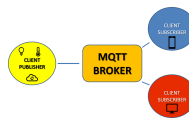
```
28 Public Sub ConnectTo(Host As String, Name As String)
29     currentName = Name
30     isServer = Host = "127.0.0.1"
31     If isServer Then
32         If brokerStarted = False Then
33             broker.Start
34             brokerStarted = True
35         End If
36         users.Clear
37         Host = "127.0.0.1"
38     End If
39     If connected Then client.Close
40     client.Initialize("client", $"tcp://{Host}:{port}", "android" & Rnd(1, 1000000))
41     Dim mo As MqttConnectOptions
42     mo.Initialize("", "")
43     'this message will be sent if the client is disconnected unexpectedly.
44     mo.SetLastWill("all/disconnect", serializer.ConvertObjectToBytes(currentName), 0, False)
45     client.Connect2(mo)
46 End Sub
47
48 Private Sub client_Connected (Success As Boolean)
49     Log($"Connected: ${Success}")
50     If Success Then
51         connected = True
52         client.Subscribe("all/#", 0)
53         client.Publish2("all/connect", serializer.ConvertObjectToBytes(currentName), 0, False)
54     Else
55         ToastMessageShow("Error connecting: " & LastException, True)
56     End If
57 End Sub
--
```



Easy TCP/IP for IoT

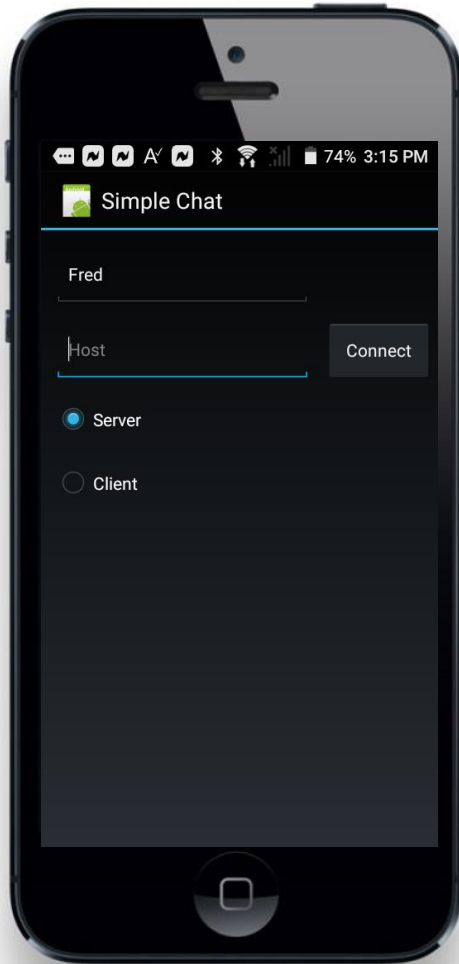
Firmware - MQTT:B4A Perspective - Embedded Broker

```
59 Private Sub client_MessageArrived (Topic As String, Payload() As Byte)
60     Dim receivedObject As Object = serializator.ConvertBytesToObject(Payload)
61     If Topic = "all/connect" Or Topic = "all/disconnect" Then
62         'new client has connected or disconnected
63         Dim newUser As String = receivedObject
64         If isServer Then
65             Log($"{Topic}: ${newUser}")
66             Dim index As Int = users.IndexOf(newUser)
67             If Topic.EndsWith("connect") And index = -1 Then users.Add(newUser)
68             If Topic.EndsWith("disconnect") And index >= 0 Then users.RemoveAt(index)
69             client.Publish2("all/users", serializator.ConvertObjectToBytes(users), 0, False)
70         End If
71     Else if Topic = "all/users" Then
72         Dim newUsers As List = receivedObject
73         CallSubDelayed2(Chat, "NewUsers", newUsers) 'this will start the chat activity if it wasn't started yet.
74     Else
75         Dim m As Message = receivedObject
76         CallSub2(Chat, "NewMessage", m)
77     End If
78
79 End Sub
```



Easy TCP/IP for IoT

Firmware - MQTT:B4A Perspective - Embedded Broker



Easy TCP/IP for IoT

Day 5 Summary

