**********Draft**********

Ultibo virtual MQTT Broker
connection with nano-rp2040-connect
with a goal to use a pico_w instead
11/28/22

**********Draft**********

Required: QEMU 6.2.0 compiled from source. qemu-6.2.0-rpios-32bit.img or qemu-6.2.0-rpios-64bit.img. Openocd is also compiled from source installed-openocd050322-610f137.img installed-openocd082722-228ede-64bit.img. Lazarus IDE (Ultibo-Edition) is installed using https://github.com/develone/Tools/blob/master/Installer/Core/Linux/ultiboinstaller.sh
The MQTT will run on a Raspberry Pi OS 32 or 64 bit system.
Arduino IDE was installed on a 32 bit Raspberry Pi OS.

nano-rp2040-connect configuration.

```
const char broker[] = "192.168.1.229"; // QEMU pi4-27 Address of the MQTT server
int        port     = 9883;

const char topic[]  = "update/memo";

mqttClient.setId("nano-rp2040-connect");

mqttClient.setUsernamePassword("testuser", "password123");

while (!mqttClient.connect(broker, port)) {
                .
                .
                .
}
```

Since the MQTT Broker is running in a virtual system the mqtt-port is 9883 instead of 1883

Goal: To use a pico_w instead of nano-rp2040-connect to connect to MQTT Broker running virtually on Raspberry Pi 4B. The code for mqtt broker was written by Ultibo user https://github.com/pjde/ultibo-mqtt which was added to https://github.com/develone/Ultibo_Projects/tree/master/Pauls-ultibo-mqtt

Status: Two files are used on the nano-rp2040-connect arduino_secrets.h & mqtt.ino plus

```
libraries/ArduinoMqttClient/
├── examples
│   ├── WiFiAdvancedCallback
│   │   ├── arduino_secrets.h
```

```
│   │   └── WiFiAdvancedCallback.ino
│   ├── WiFiEcho
│   │   ├── arduino_secrets.h
│   │   └── WiFiEcho.ino
│   ├── WiFiEchoCallback
│   │   ├── arduino_secrets.h
│   │   └── WiFiEchoCallback.ino
│   ├── WiFiSimpleReceive
│   │   ├── arduino_secrets.h
│   │   └── WiFiSimpleReceive.ino
│   ├── WiFiSimpleReceiveCallback
│   │   ├── arduino_secrets.h
│   │   └── WiFiSimpleReceiveCallback.ino
│   └── WiFiSimpleSender
│       ├── arduino_secrets.h
│       └── WiFiSimpleSender.ino
├── keywords.txt
├── library.properties
├── LICENSE.txt
├── README.adoc
└── src
    ├── ArduinoMqttClient.h
    ├── MqttClient.cpp
    └── MqttClient.h
```

8 directories, 19 files

The current pico_w code is found at https://github.com/develone/pico_w-mqtt_example
This code compile links and executes but does not connect like nano-rp2040-connect.

pico_w configuration.

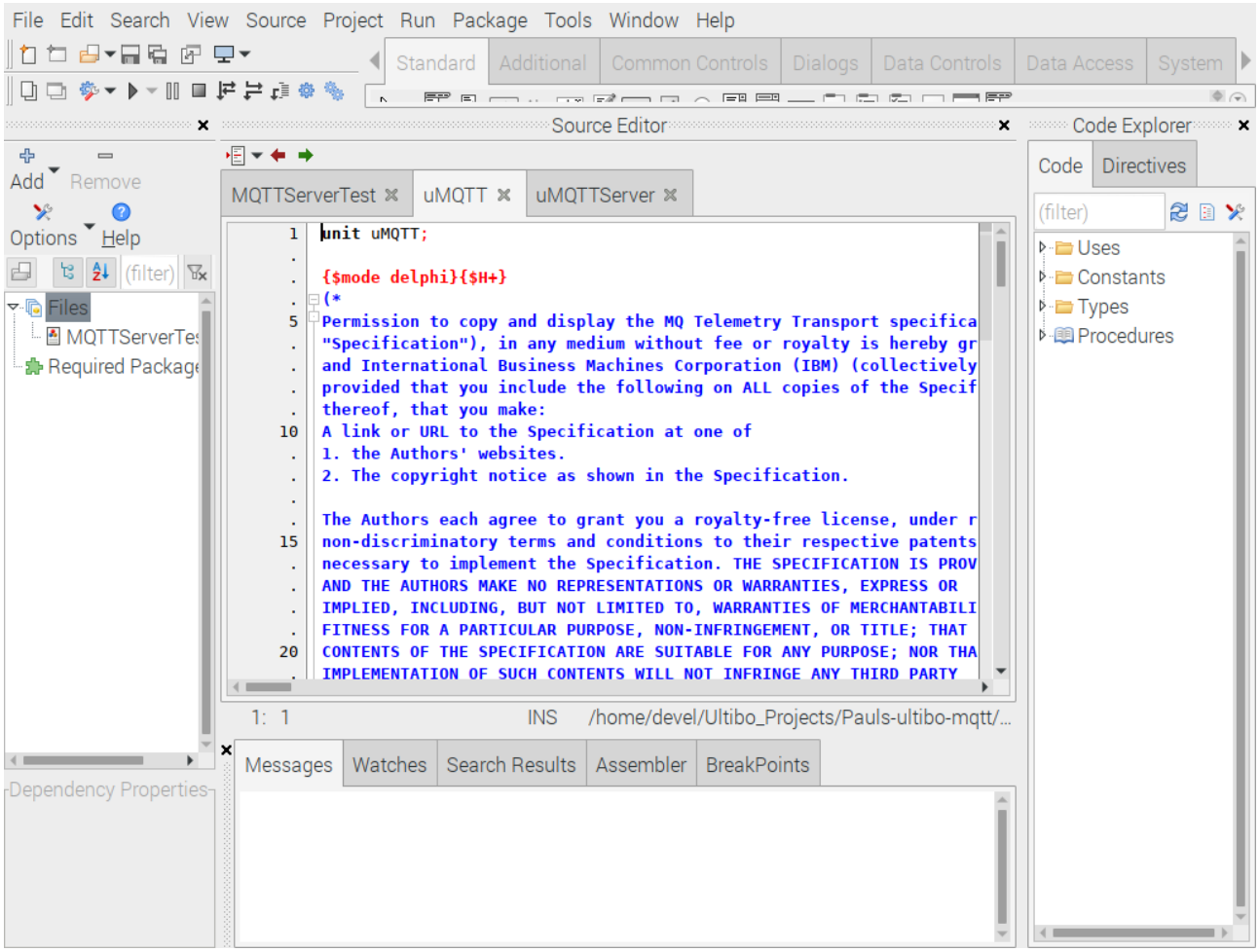
u16_t mqtt_port = 9883;

#define LWIP_MQTT_EXAMPLE_IPADDR_INIT = IPADDR4_INIT(PP_HTONL(0xc0a801e5))

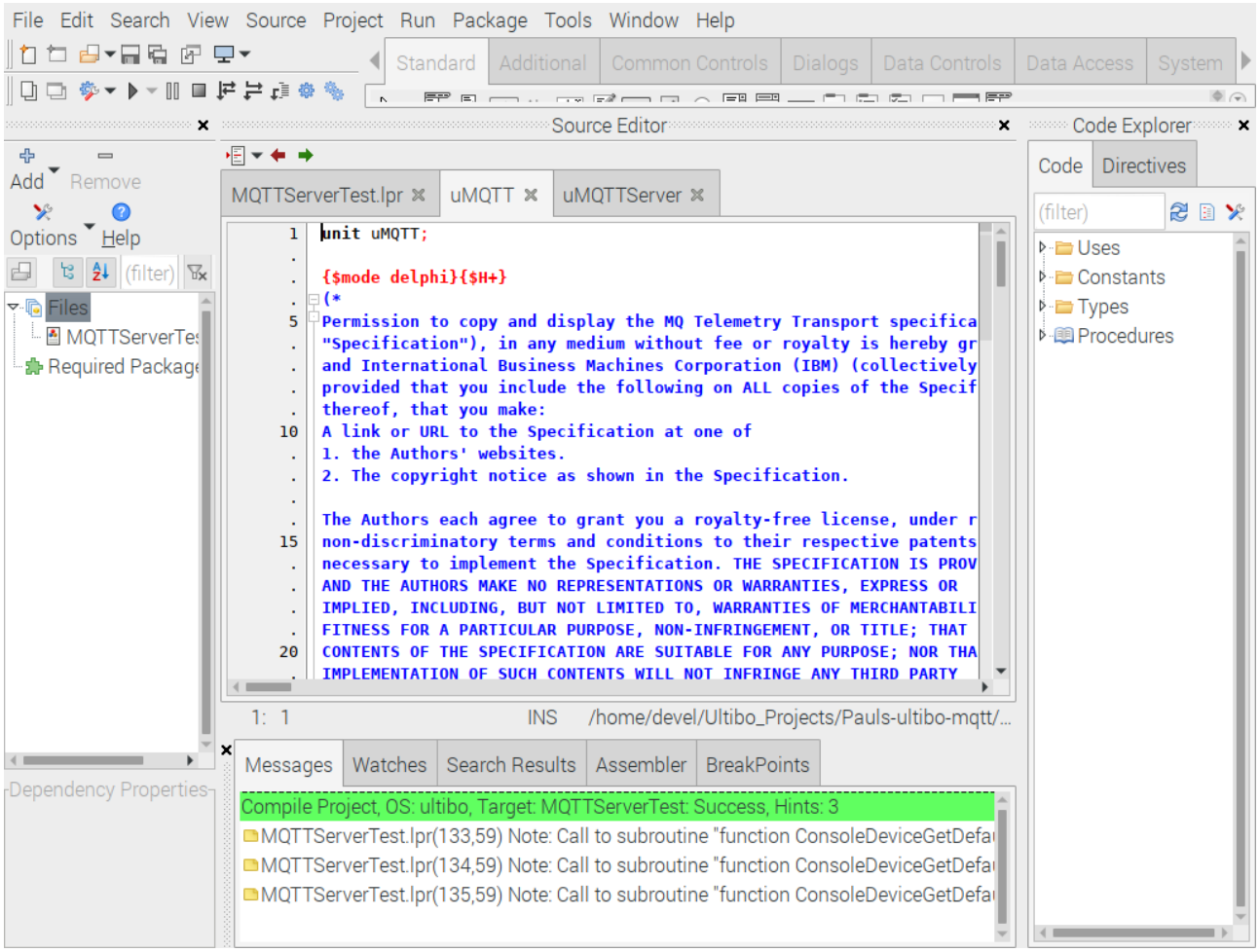
The following command is used to set the path for openocd & qemu.

. Ultibo_Projects/picoultibo.sh

```
export ULTIBO1=/home/devel/ultibo/core
export ULTIBO2=/home/devel/qemu-6.2.0-rpios/bin
export PICO=/home/devel/local/openocd/bin
export PICOTOOL=/home/devel/picotool/build/
export PATH=$ULTIBO1:$ULTIBO2:$PICO:$PICOTOOL:$PATH
echo $PATH
```

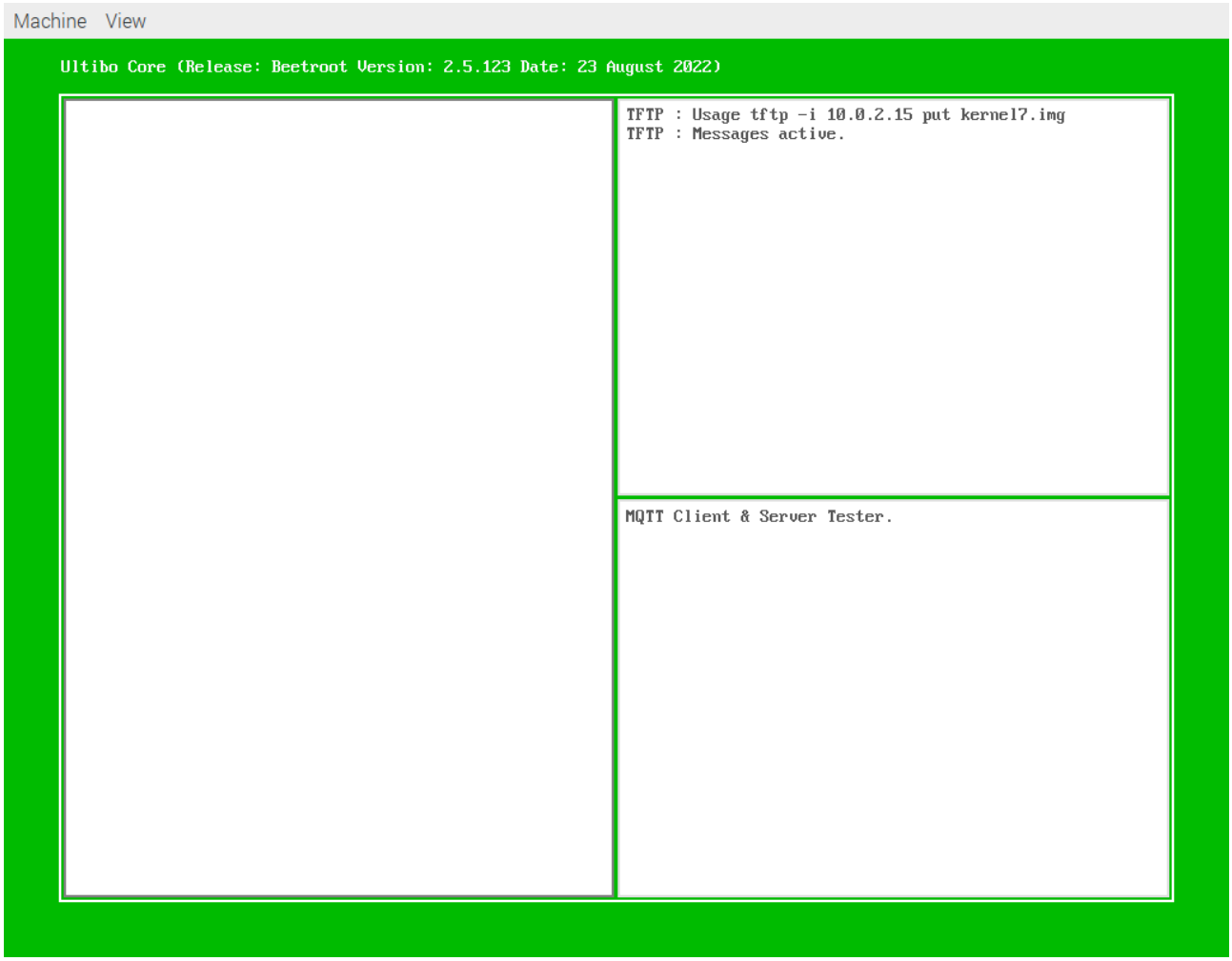


From the main menu Run/Compile will generate a “kernel.bin” when the green bar appears.



The script startqemu.sh will bring up the MQTT broker.

```
#!/bin/bash
qemu-system-arm -machine versatilepb -cpu cortex-a8 -kernel kernel.bin \
-net
user,hostfwd=tcp::5080-:80,hostfwd=tcp::5023-:23,hostfwd=tcp::9883-:1883,hostfwd=udp::5069-:6
9,hostfwd=tcp::6050-:5050 -net nic \
 -drive file=disk.img,if=sd,format=raw
```



When a 5 is depressed in the left panel currently the nano-rp2040-connect and begin sending messages. The nano-rp2040 will connect to the mqtt broker as client "nano-rp2040-connect" user "testuser" password "password123" and start sending messages on topic "update/memo".

cat /dev/ttyACM0
Sent MQTT message.

Received a message with topic 'update/memo', length 10 bytes:

BUTTON 399

Sent MQTT message.

Received a message with topic 'update/memo', length 10 bytes:

BUTTON 400

Sent MQTT message.

Received a message with topic 'update/memo', length 10 bytes:

BUTTON 401

Sent MQTT message.

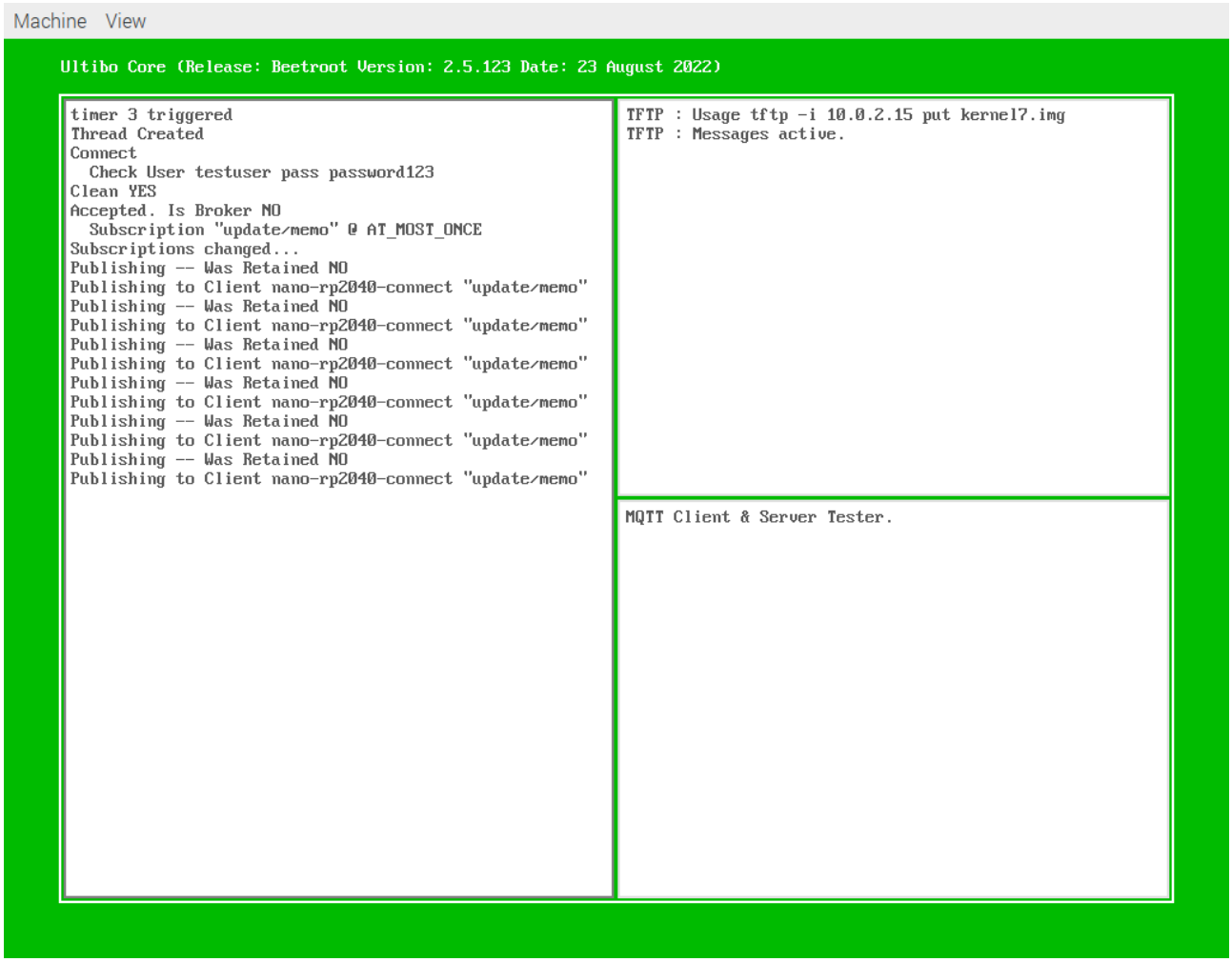
Received a message with topic 'update/memo', length 10 bytes:

BUTTON 402

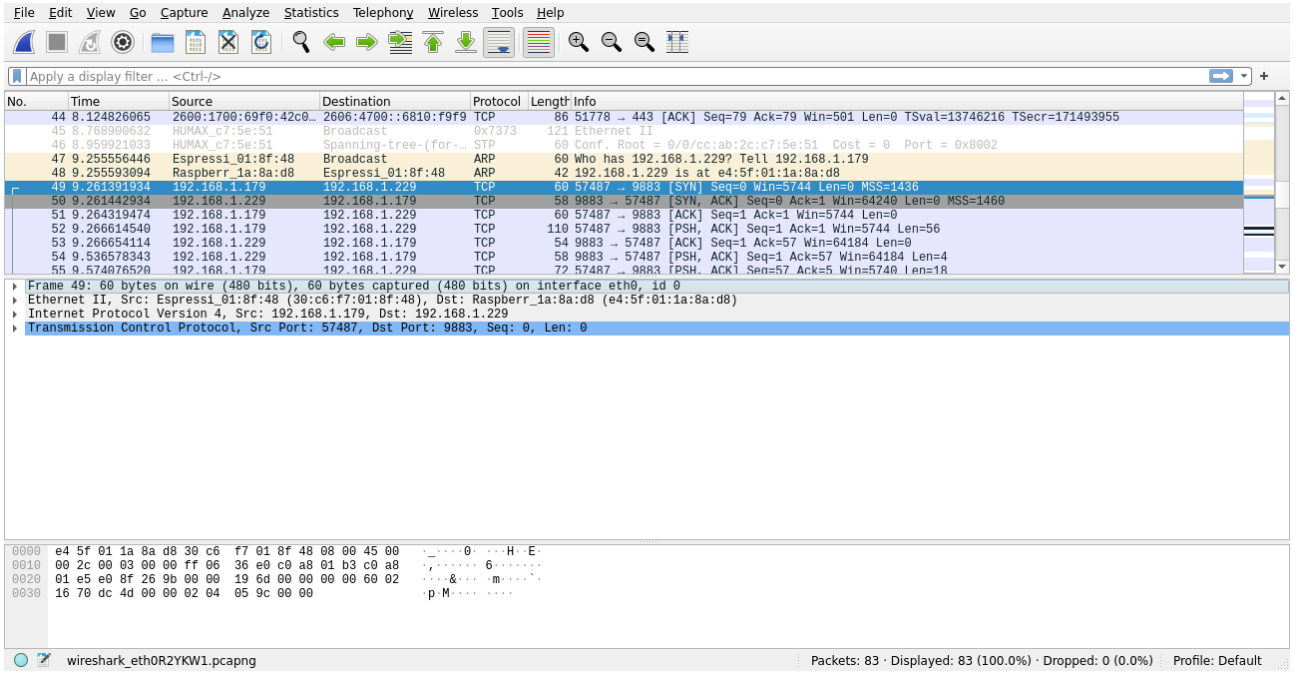
Sent MQTT message.

Received a message with topic 'update/memo', length 10 bytes:

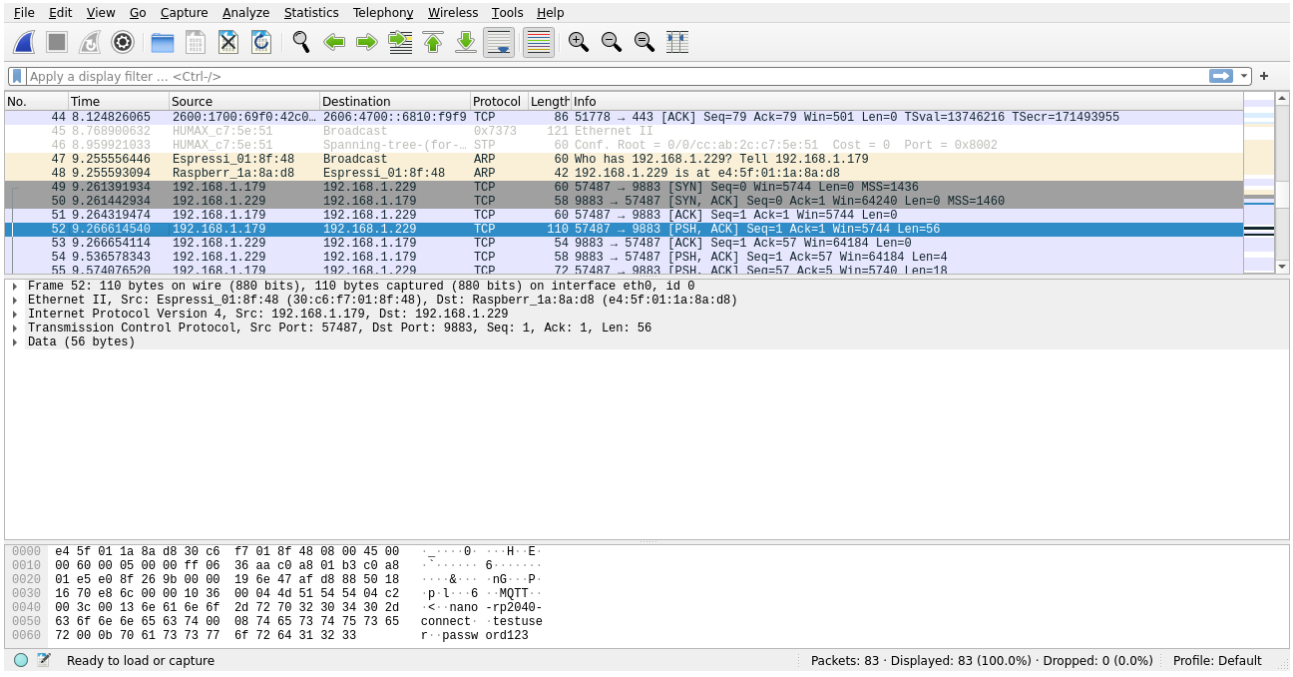
BUTTON 403



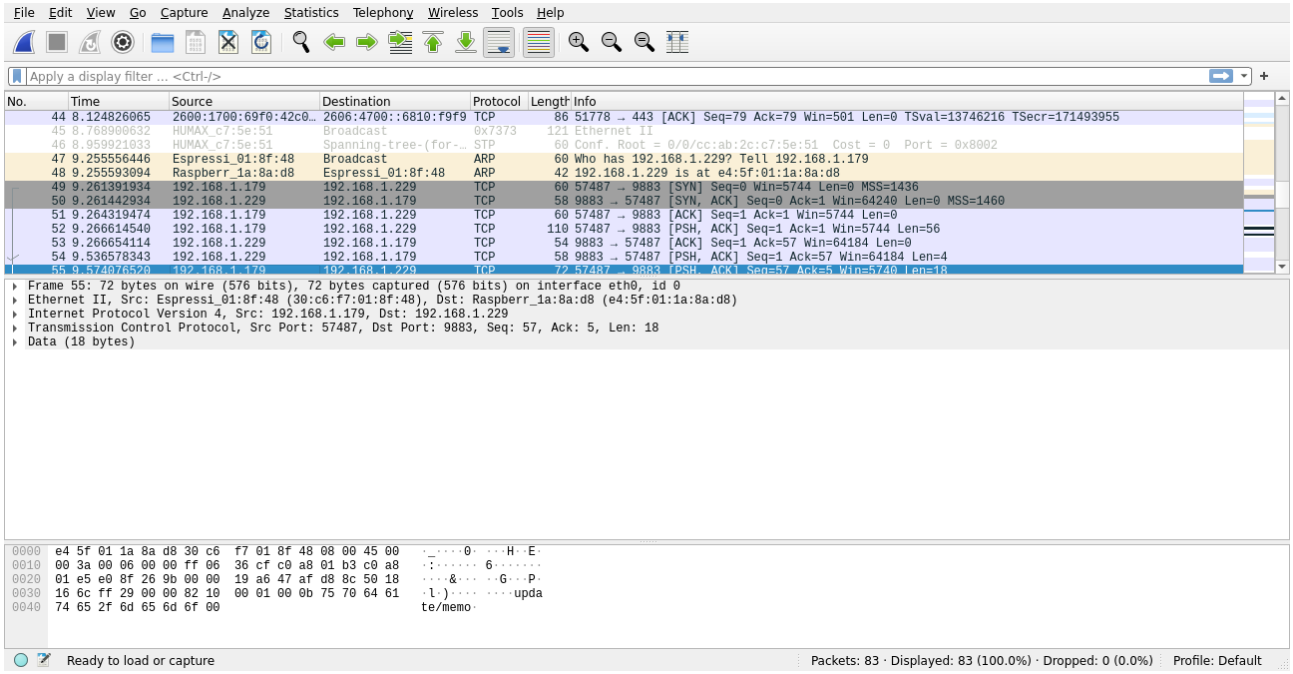
The nano-rp2040-connect



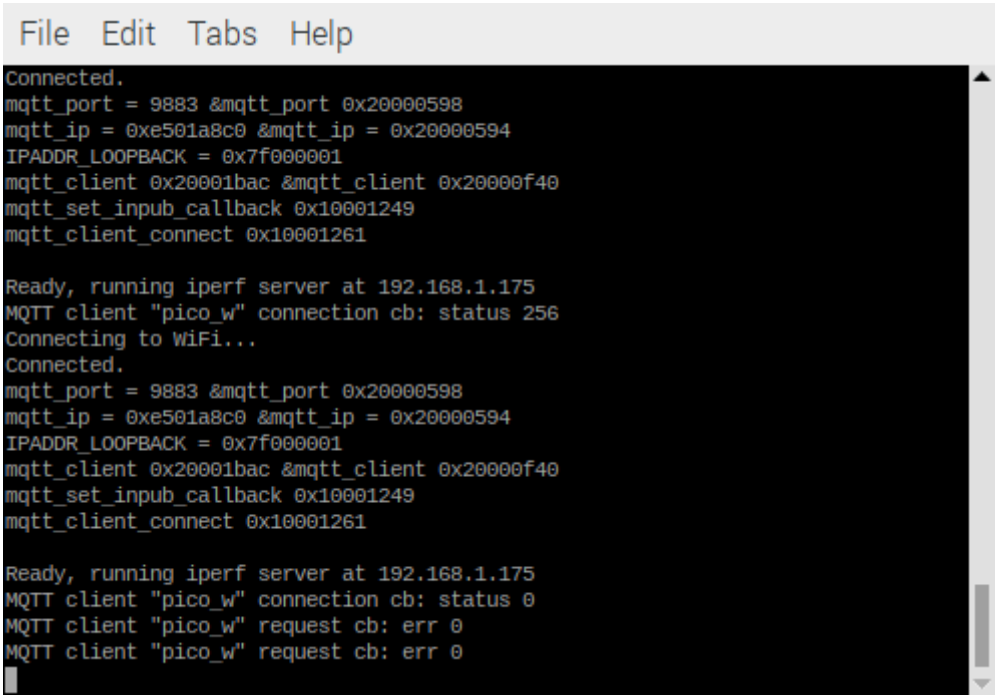
This when the connection information is sent from the nano-rp2040-connect to the mqtt broker.



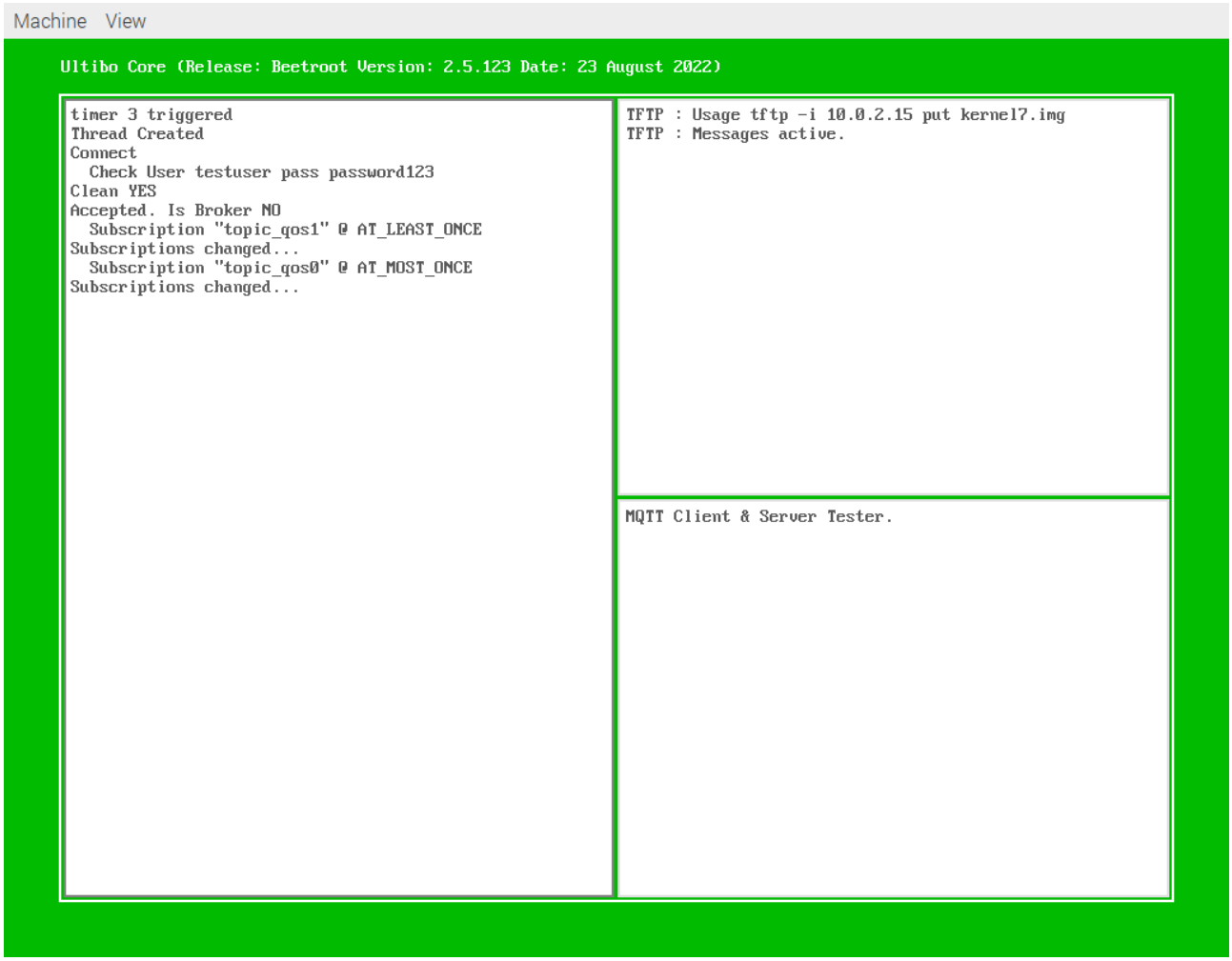
The nano-rp2040-connect begins sending updates to topic “update/memo” to mqtt broker.



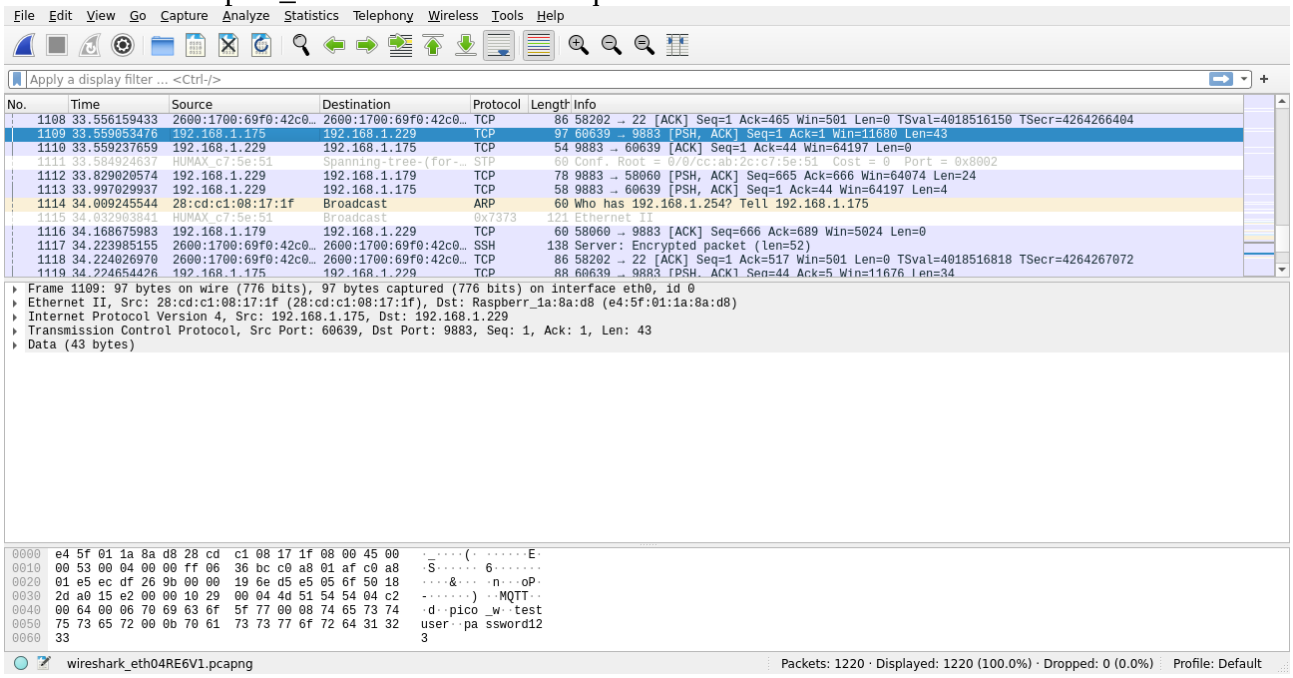
pico_w connects with MQTT Broker



pico_w connects with MQTT Broker



This Is when the pico_w connects with the mqtt broker.



Starting FreeRTOS on core 0:
Connecting to WiFi...

Connected.
Connecting to WiFi...
Connected.
mqtt_port = 9883 &mqtt_port 0x20000598
mqtt_ip = 0xe501a8c0 &mqtt_ip = 0x20000594
IPADDR_LOOPBACK = 0x7f000001
mqtt_client 0x20001bac &mqtt_client 0x20000f40
mqtt_set_inpub_callback 0x10001249
mqtt_client_connect 0x10001261

Ready, running iperf server at 192.168.1.175
MQTT client "pico_w" connection cb: status 0
MQTT client "pico_w" request cb: err 0
MQTT client "pico_w" request cb: err 0