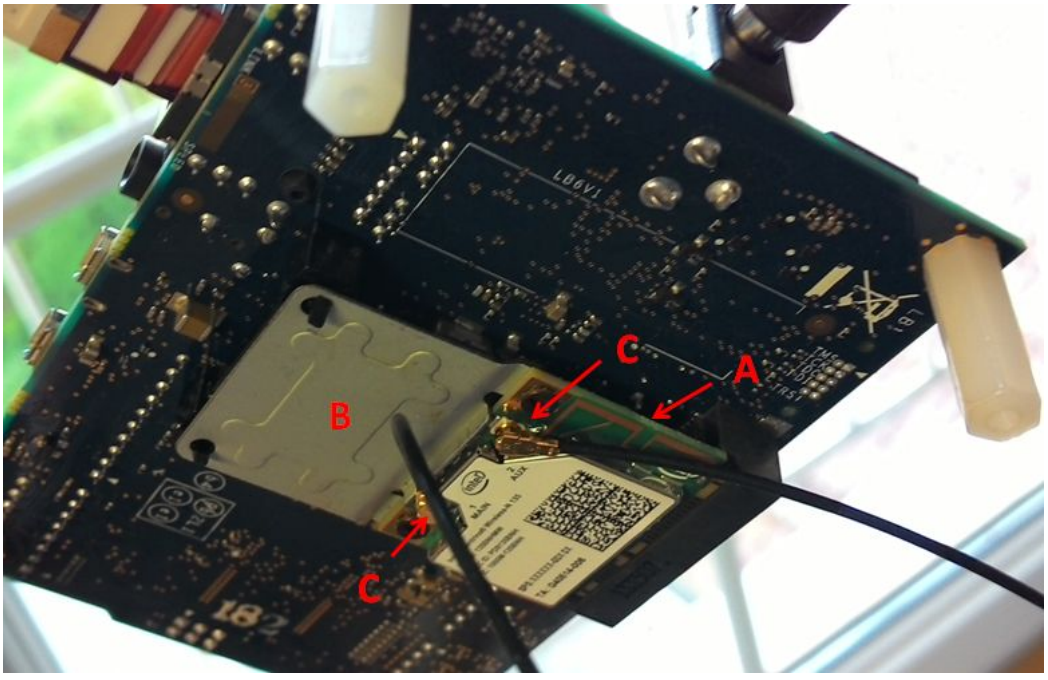
WiFi access on Intel® Galileo with Yocto* Linux

Submitted by Matthias H. (Intel) (https://software.intel.com/en-us/user/183576) on April 25, 2014

Install wireless Module

Intel® Galileo comes with a full length mini PCIe slot which is shown in the picture below fitted with an Intel® Centrino™ Wireless-N 135 adapter (A). If you have a half size adapter like in the image you'll have to use a half to full size adapter (B) in order to properly mount the adapter. Finally put antennas in the jacks (C).



We assume you do have a Yocto* Linux either built from http://git.yoctoproject.org/cgit/cgit.cgi/meta-intel-iot-devkit/ or downloaded from https://software.intel.com/en-us/iotdevkit on a SD card.

Once the system is powered up you should find the adapter

```
# lspci -k | grep -A 3 -i "network"
01:00.0 Network controller: Intel Corporation Centrino Wireless-N 135 (rev c4)
                Subsystem: Intel Corporation Centrino Wireless-N 135 BGN
                Kernel driver in use: iwlwifi
                Kernel modules: iwlwifi
```

Connman

The Intel® IoT devkit Yocto* contains the embedded connection manager *connman*. We assume you also have the package *connman-client* installed which provides the *connmanctl* command line tool.

Connman can be thought of as a frontend to various other programs like *rfkill*, and *wpa_supplicant*. I.E. connman supports e.g. the rfkill (un-)block

for wifi modules such that you don't have to call rfkill directly.

```
# connmanctl

connmanctl> enable wifi

    Enabled wifi


connmanctl> technologies

    /net/connman/technology/ethernet

      Name = Wired

      Type = ethernet

      Powered = True

      Connected = True

      Tethering = False

    /net/connman/technology/bluetooth

      Name = Bluetooth

      Type = bluetooth

      Powered = True

      Connected = False

      Tethering = False

    /net/connman/technology/wifi

      Name = WiFi

      Type = wifi

      Powered = True

      Connected = False

      Tethering = False

      TetheringIdentifier = Galileo

      TetheringPassphrase = passphrase


connmanctl> scan wifi

    Scan completed for wifi


connmanctl> services

    *AO Wired                ethernet_000000000000_cable

        <WiFi SSID1>             wifi_<adapter MAC>_<hotspot 1 numeric SSID>_<type e.g. managed>_<enc

        <WiFi SSID2>             wifi_<adapter MAC>_<hotspot 2 numeric SSID>_<type e.g. managed>_<enc

  [...]
```

Let's switch auto connect and dhcp for ipv4 on:

```
connmanctl> config wifi_<MAC_a>_<MAC_h>_managed_psk --autoconnect yes --ipv4 dhcp

    Error wifi_<MAC_a>_<MAC_h>_managed_psk: Invalid service

    Service      wifi_<MAC_a>_<MAC_h>_managed_psk Ethernet = [ Method=auto, Interface=wlp1s0, Address

    Service      wifi_<MAC_a>_<MAC_h>_managed_psk IPv4.Configuration = [ Method=dhcp ]
```

In order to connect to a secured AP we need to switch the agent on which allows user interaction on the console:

```


connmanctl> agent on

    Agent registered

connmanctl> connect wifi_0cd2926de3ae_486f6d65574c414e_managed_psk

    Agent RequestInput wifi_0cd2926de3ae_486f6d65574c414e_managed_psk

      Passphrase = [ Type=psk, Requirement=mandatory ]

    Connected wifi_0cd2926de3ae_486f6d65574c414e_managed_psk

    Passphrase? <passphrase goes here>
```

Now we can check whether the connection is fine. You should find a /var/lib/connman/wifi_<MAC_a>_<MAC_h>_managed_psk/settings file and should see wifi connected fine:

```
connmanctl> technologies
    /net/connman/technology/ethernet
      Name = Wired
      Type = ethernet
      Powered = True
      Connected = True
      Tethering = False
    /net/connman/technology/bluetooth
      Name = Bluetooth
      Type = bluetooth
      Powered = True
      Connected = False
      Tethering = False
    /net/connman/technology/wifi
      Name = WiFi
      Type = wifi
      Powered = True
      Connected = True
      Tethering = False
      TetheringIdentifier = Galileo
      TetheringPassphrase = passphrase
connmanctl> services
*AO Wired                ethernet_000000000000_cable
    *AR <WiFi SSID1>             wifi_<adapter MAC>_<hotspot 1 MAC>_<type e.g. managed>_<encryption e
           <WiFi SSID2>             wifi_<adapter MAC>_<hotspot 2 MAC>_<type e.g. managed>_<encryptio



# cat /var/lib/connman/wifi_<MAC_a>_<MAC_h>_managed_psk/settings
[wifi_<MAC_a>_<MAC_h>_managed_psk]
    Name=<WiFi SSID1>
    SSID=<hex representation>
    Frequency=2462
    Favorite=true
    AutoConnect=true
    Modified=2014-05-17T14:11:31.017109Z
    Passphrase=<passphrase in plain text>
    IPv4.method=dhcp
    IPv4.DHCP.LastAddress=<IP address>
    IPv6.method=auto
    IPv6.privacy=disabled
connmanctl> services  wifi_<MAC_a>_<MAC_h>_managed_psk
    /net/connman/service/wifi_<MAC_a>_<MAC_h>_managed_psk
      Type = wifi
      Security = [ psk ]
      State = ready
      Strength = 78
      Favorite = True
      Immutable = False
```

```
099       AutoConnect = True
100
101       Name = <SSID>
102
103       Ethernet = [ Method=auto, Interface=wlp1s0, Address=<MAC>, MTU=1500 ]
104
105       IPv4 = [ Method=dhcp, Address=<IP>, Netmask=<MASK> ]
106
107       IPv4.Configuration = [ Method=dhcp ]
108
109       IPv6 = [  ]
110
111       IPv6.Configuration = [ Method=auto, Privacy=disabled ]
112
113       Nameservers = [ <IP> ]
114
115       Nameservers.Configuration = [  ]
116
117       Timeservers = [ <IP> ]
118
119       Timeservers.Configuration = [  ]
120
121       Domains = [ <your domain> ]
122
123       Domains.Configuration = [  ]
124
125       Proxy = [ Method=direct ]
126
127       Proxy.Configuration = [  ]
128
129       Provider = [  ]
130
131
132
133 # iwconfig
134
135     lo        no wireless extensions.
136
137 wlp1s0    IEEE 802.11bgn  ESSID:"<SSID>"
138
139               Mode:Managed  Frequency:2.462 GHz  Access Point: <MAC>
140
141               Bit Rate=1 Mb/s   Tx-Power=15 dBm
142
143               Retry  long limit:7   RTS thr:off   Fragment thr:off
144
145               Encryption key:off
146
147               Power Management:off
148
149               Link Quality=54/70  Signal level=-56 dBm
150
151               Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
152
153               Tx excessive retries:0  Invalid misc:104   Missed beacon:0
154
155 enp0s20f6  no wireless extensions.
```

Next time you boot connman will automatically reconnect to your access point

Note: If you maintain both a wired and WiFi connection at reboot, only the wired connection will be established. If you removed the wired connection, then the WiFi will attempt to connect. Reinserting the wired connection will not change this connection model, unless WiFi drops.

Categories: Internet of Things (/en-us/search/site/field_platform/internet_of_things-45744/language/en) , Embedded (/en-us/search/site/field_hardware/embedded-45747/language/en) , Intel® Galileo platform (/en-us/search/site/field_hardware/intel_galileo_platform-80495/language/en) , Developers (/en-us/search/site/field_audience/developers-17152/language/en) , Professional (/en-us/search/site/field_audience/professional-81559/language/en) , Students (/en-us/search/site/field_audience/students-17155/language/en) , Linux* (/en-us/search/site/field_operating_system/linux-20787/language/en) , Yocto Project (/en-us/search/site/field_operating_system/yocto_project-41420/language/en)

Tags: Intel® IoT Developer Kit (/en-us/search/site/field_tags/intel_iot_developer_kit-81407/language/en)

Comments (3)

Top

Ivan D. said on Thu, 11/19/2015 - 16:21

Hmm, I've been trying to use this guide to connect to an enterprise 802.1x network but I think I don't have the connmanctl syntax right. For instance, I have a config file provided by the networks team, the network is called EDU so I named it that. If I try to connect to the network it fails with errors that suggest to me I'm not driving connmanctl correctly. For instance:

connmanctl> services

wifi_001500a43360_hidden_managed_ieee8021x

EDU wifi_001500a43360_454455_managed_ieee8021x

connmanctl> agent on

Agent registered

connmanctl> connect wifi_001500a43360_454455_managed_ieee8021x

Error /net/connman/service/wifi_001500a43360_454455_managed_ieee8021x: Invalid arguments

I also tried to follow the suggestion at the bottom of the page for an enterprise network, by creating the aforementioned settings file and trying to connect but I still get an error.

connmanctl> connect EDU

Error /net/connman/service/EDU: Method "Connect" with signature "" on interface "net.connman.Service" doesn't exist

I'm not very familiar with connmanctl so would love some help. I've now been trying to connect this Galileo board to the school network for over three months! (I only visit every fortnight).

Thanks.

Matthias H. (Intel) said on Fri, 06/20/2014 - 02:48

Note:

for a ieee8021x network you can't use the connman agent to enter your credentials but you'd rather have to manually add like shown below for a peap hotspot. After this has been added you can simply

```
connmanctl connect <WiFi service name>
```

in order to connect to the hotspot.

Configuration:

```
# cat /var/lib/connman/<just choose a nice name - important is that the name below matches>.config
[global]
Name = <hotspot name>
Description = <add whatever you like here>
[service_peap]
Type = wifi
Name = <hotspot name - same as above>
EAP = peap
Phase2 = MSCHAPV2
Identity = <user identity>
Passphrase = <secret passphrase>
```

lydia B. said on Thu, 06/19/2014 - 02:42

No comment.

Add a Comment

Top

(For technical discussions visit our developer forums. For site or software product issues contact support.)

Please sign in to add a comment. Not a member? Join today ›