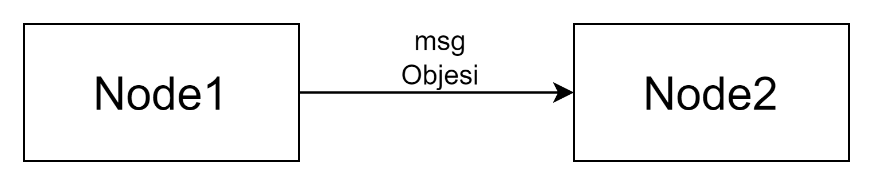
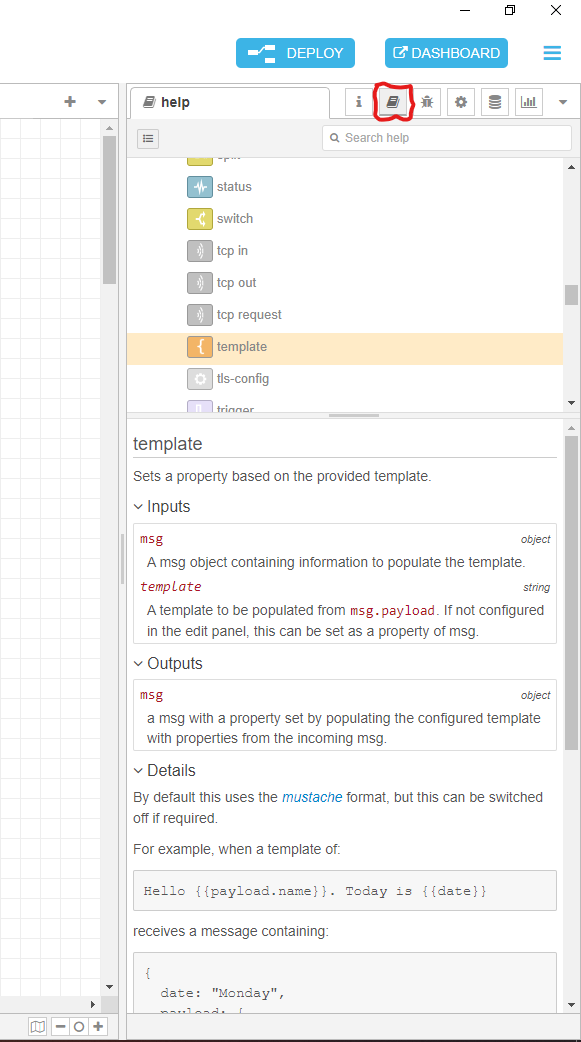
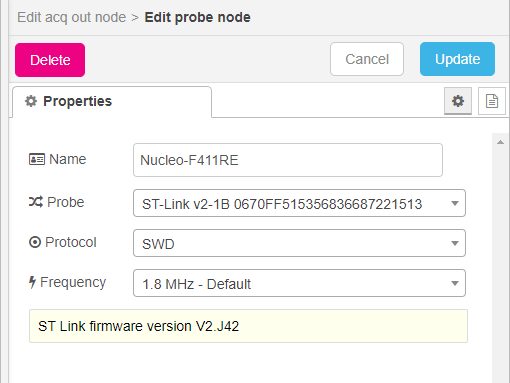
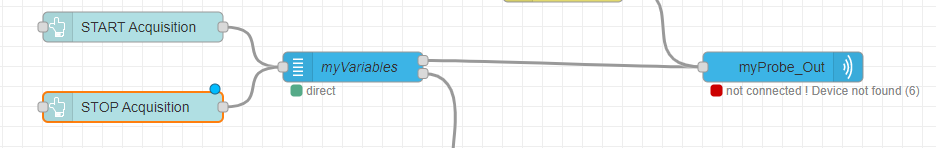
STM32CubeMonitor Led Example

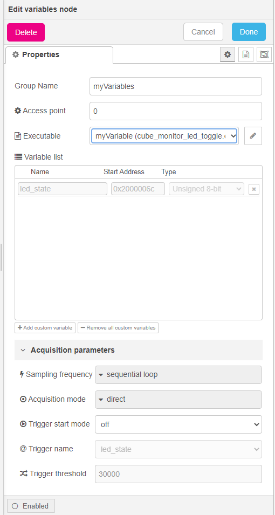
Öncelikle STM32CubeMonitor’de bulunan Node’ların ne yaptıklarını ve hangi tip msg.topic ve msg.payload aldıklarını veya çıktı verdiklerini sidebar’da kitap işaretine bastıktan sonra Node’ların üstüne bakarak görebilirsiniz.

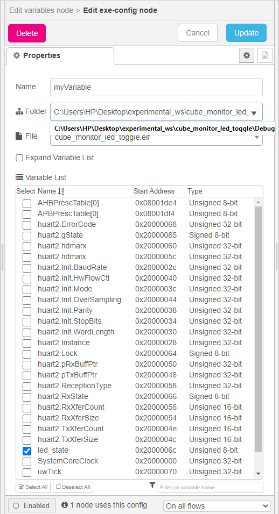
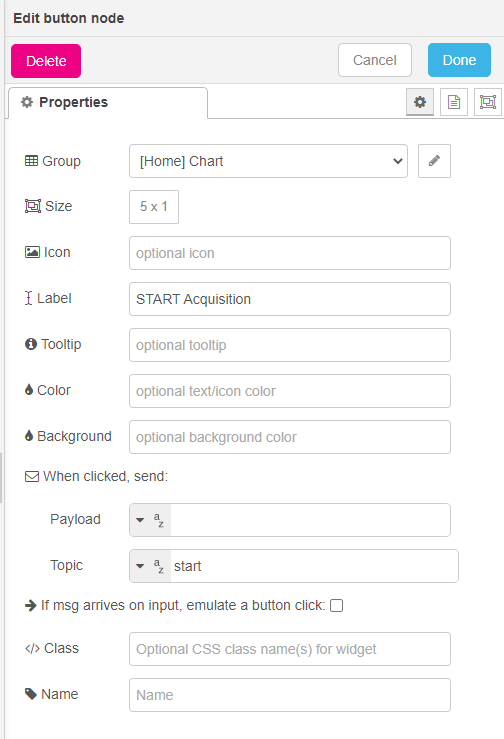
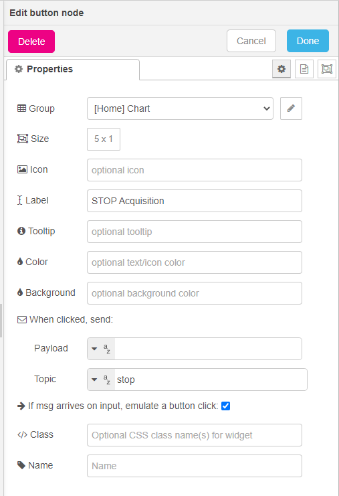
CubeMonitor’de bir Node başka bir Node ile msg objesi kullanark haberleşmektedir. Msg objesinin msg.topic ve msg.payload olamk üzere 2 parçası bulunmaktadır. msg.topic genelde girdiği Node’a göre yapılacak işlemi seçerken msg.payload değişken bilgilerini içermektedir.

Projenin Başlangıcında Yapılması Gereken Ayarlar

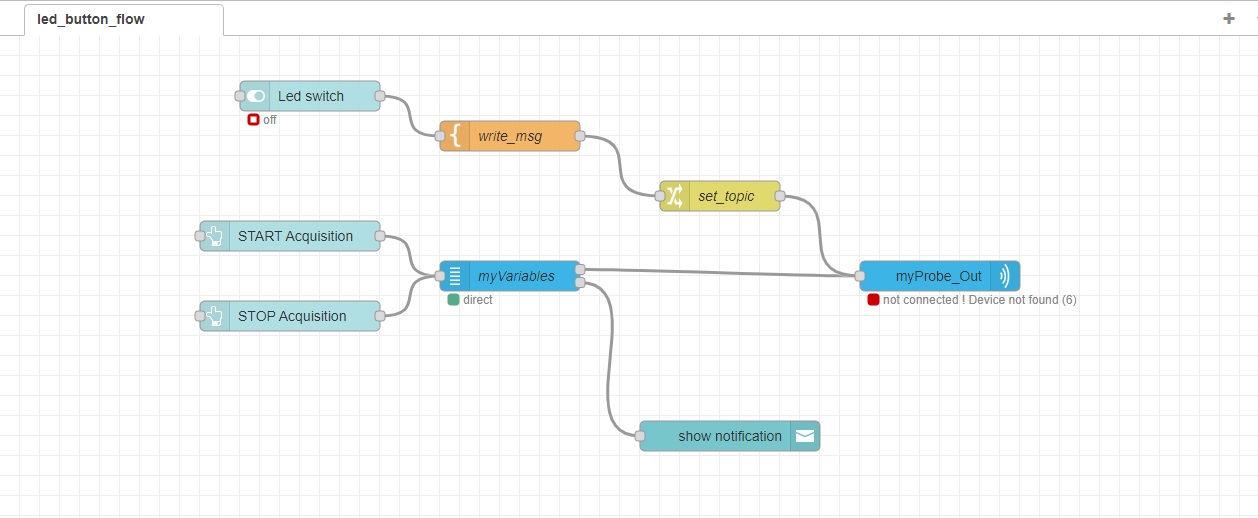
“acq out” ya da “acq in” kullanıyorsanız sistemin çalışması için bu Node’ların ayarlarından kullandığınız ST-Link’in eklenmesi gerekmektedir. Ardından bu Node’lara systemin başlatılması için “start”, sistemin durdurulması için “stop” msg.topic verilmesi gerekmektedir.



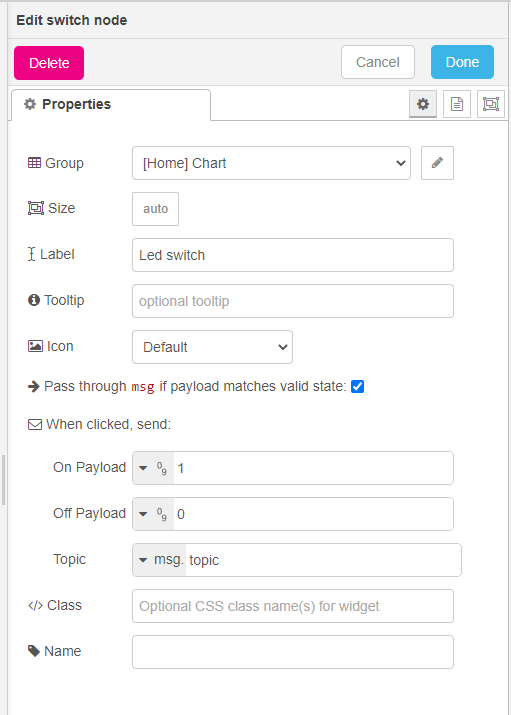
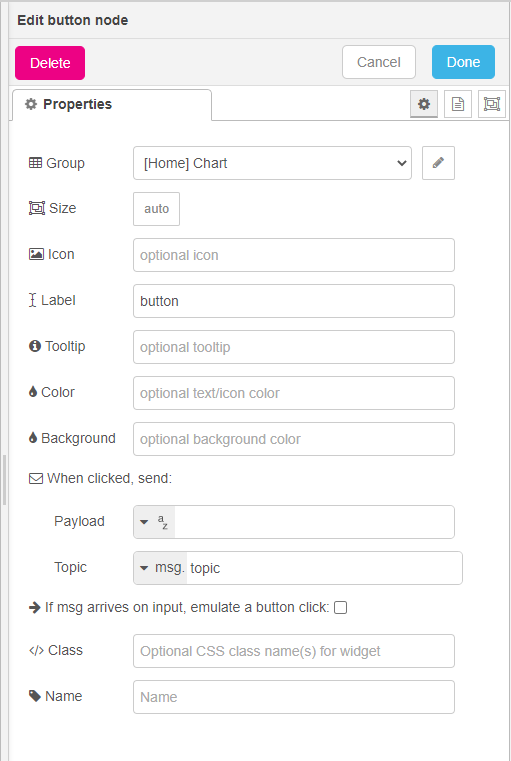
“variables” Node’undan da kullanılacak değişkenleri seçilim “Variable List”e eklenmesi gerekmektedir. Kartınızda kullandığınız değişkenleri bu listede görebilmeniz için yüklediniz projenin “.elf” dosyasını (projenin Debug klasöründe bulabilirsiniz) Node ayarlarından eklemeniz gerekmektedir. Aşağıda buton ayarları ile birlikte bu ayarlar görülebilir.

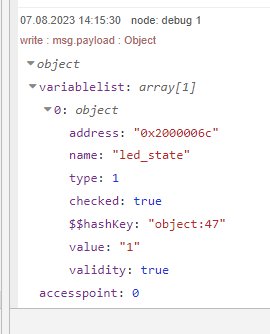


Kullanılan sistemin Node yapısı aşağıda gösterilmiştir.



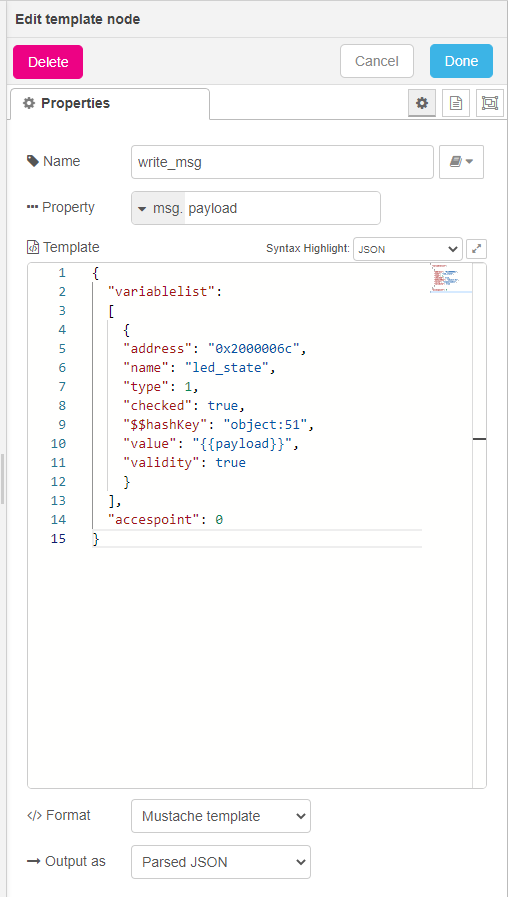
Öncelikle normal “buton” Node’nun kullanılmamasının sebebi normal “buton” Node’u sadece butona basıldığında bir tip msg objesi atabilmektedir. Kullanılan “switch” Node’unun ise ON ve OFF olduğu duruma göre gönderdiği değer farklıdır. Aşağıda switch’in ayarları ile normal butonun ayarlarının karşılaştırılması gösterilmektedir.



Sistemde aslında yapılan şey “write panel” Nodu’nun istenilen değişkende bir değişiklik yapıldığında yolladığı msg objesinin taklit edilmesidir. Mesajın örneği sağda görülebilir.

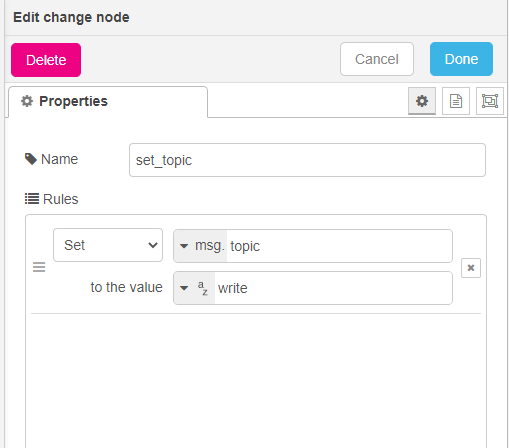
Bu görüntü “debug” Node’unun “write panel” Node’unun çıkışına bağlanarak debug ekranında gönderilen mesajın görüntüsünü çıktı vermesi sayesinde alınmıştır.

Bu mesajın taklidini yapabilmek için de “template” Node’u kullanılmıştır. Node içeriği ve ayarları aşağıda görülebilir.



“Syntax highlight”tan yazıldığı formata göre düzeni görülebilir. Object çıktısı vermesi için çıktının Parsed JSON olması gerekir yoksa direk text olarak çıktı vermektedir. Switchden gelen veri direct olarak payload olup gelen veriye göre doldurma “{{payload}}” olarak yapılır. Başka değişkenler için de aynı yazım sitili kullanılabilir.

Bu “template”den çıkan mesaj topic olarak eksiktir. Örnek aldığımız mesajın ilk satırında “write : msg payload : Object” yazısında “write” msg.topiciçeriğidir ve bunun eklenmesi gerekmektedir. Bu da “change” Node’u ile oluşturduğumuz taklit mesajına eklenir. “change” Node’unun ayarları aşağıdaki gibidir.



PS: Help ekranında Node’lar ile ilgili verilen bilgilerden bağzılarının çıkışları yazanla aynı değil. Bu durum döküman yanlışlığından mı yoksa Json standard bunlar olduğu için mi bilmemekteyim.

[Örnek\_Hazırlanırkan\_Kullanılan\_Wiki](https://wiki.st.com/stm32mcu/wiki/STM32CubeMonitor:How_to_write_a_variable_from_flow)

[Tutorial\_Videolarının\_Başlangıcı](https://www.youtube.com/watch?v=YWjvHhgqvm4)

[STM32CubeMonitor\_Wiki](https://wiki.st.com/stm32mcu/wiki/Category:STM32CubeMonitor_tools_suite)

[Mustache\_Manual](http://mustache.github.io/mustache.5.html)