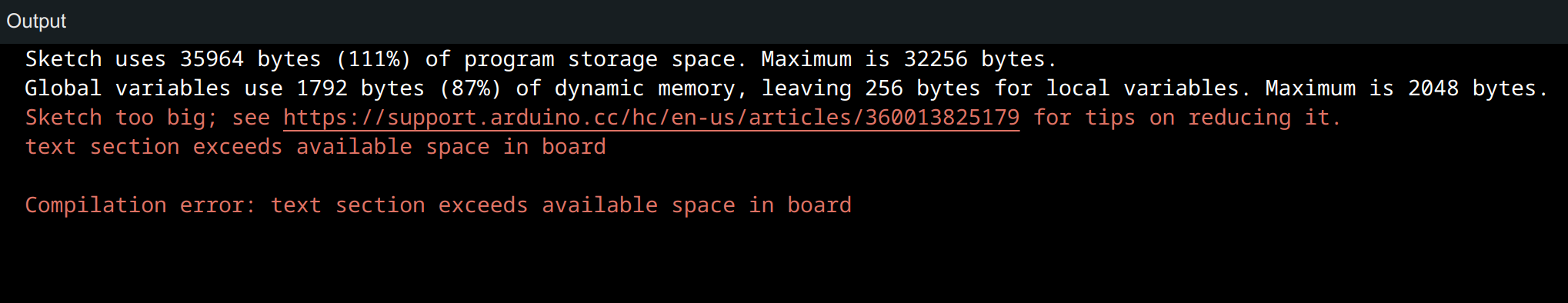
BI-ARD Semestrální práce – Smart sensor teploty a vlhkosti

# Zpráva

Semestrální práce na Arduino mě bavila, a nakonec jsem se dostal k funkčnímu řešení, s kterým jsem relativně spokojený. To ale neznamená, že jsem nenarazil na žádné zádrhele. Jak už jsem předpověděl v teoretické přípravě, narazil jsem na problém, že Arduino Uno prostě není na toto moje použití vhodné, a to primárně z důvodu nedostatku paměti a úložiště (pouze 2KB + 32KB).

Vzhledem k tomu, že jsem chtěl provozovat web sever, už ze začátku jsem potřeboval docela dost knihoven, abych to celé mohl zprovoznil. Chtěl jsem zároveň integrovat možnost logování na SD kartu a připojení k MQTT serveru. To se mi nakonec nepovedlo. Pro hostování webserveru jsem potřeboval mít v kódu hardcoded aspoň nějaké základní HTML, a jak jsem zjistil stringy zabírají nejvíce paměti RAM. Někdy se mi i povedlo program zkompilovat, ale narazil jsem na korupci dat nebo crashe.

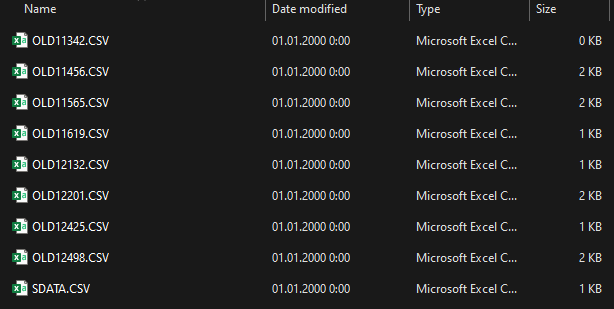


Řešení, které jsem nakonec zvolil, je nechat uživatele prostřednictvím definů v kódu zvolit, jestli chce používat logování na SD kartu, nebo přes MQTT do Home Assistenta. Obě dvě verze fungují, přičemž ve funkčnosti se mění pouze způsob ukládání dat.

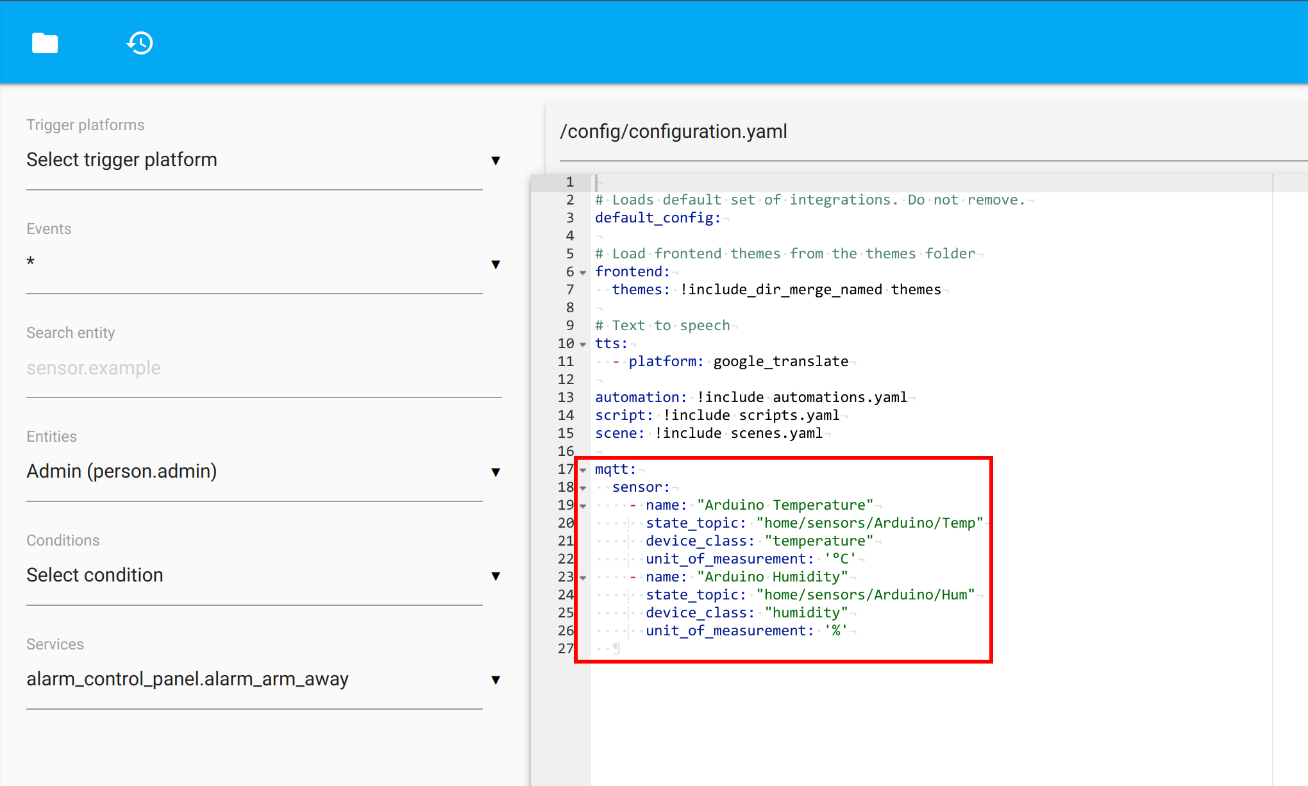
## Logování na SD kartu

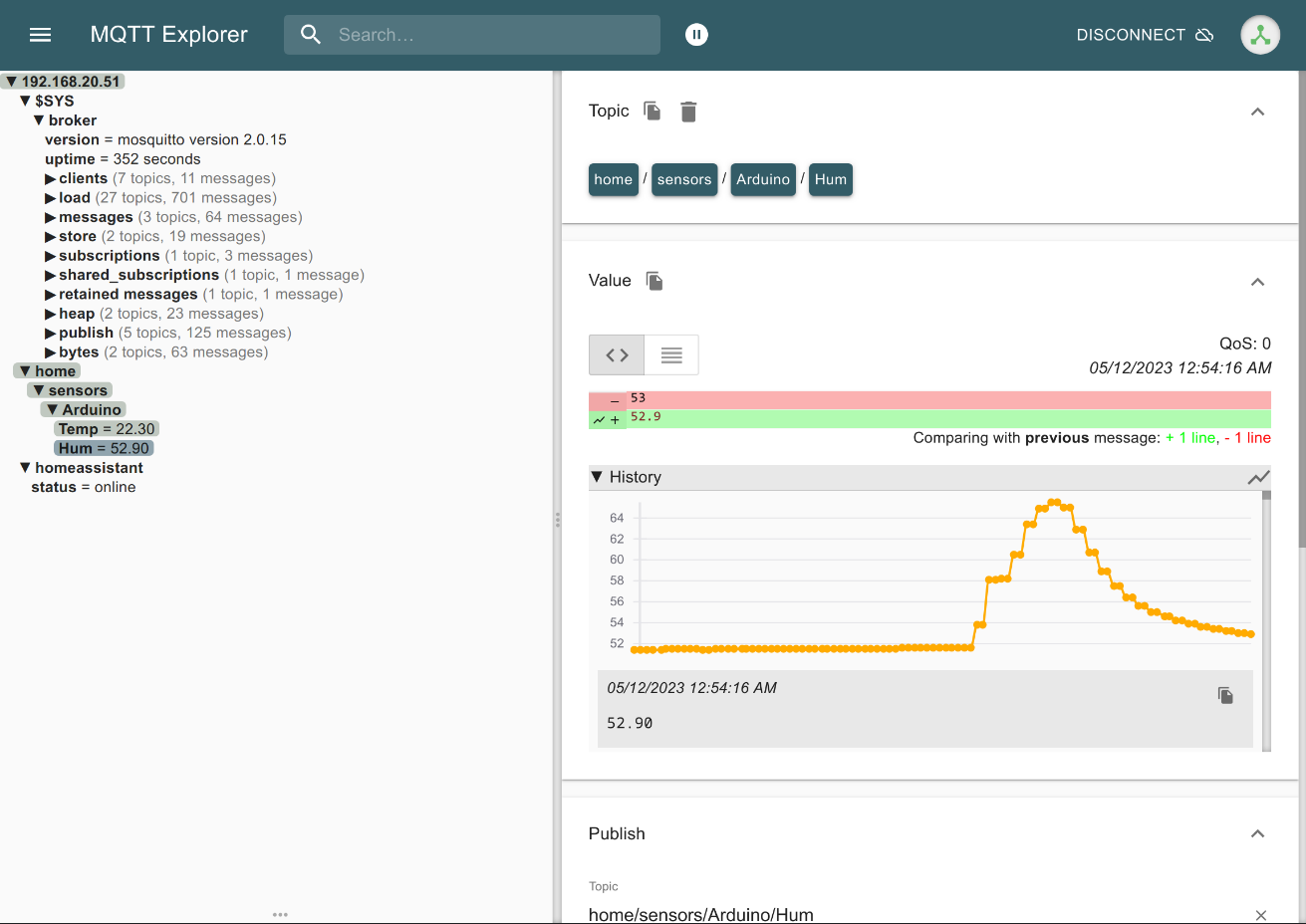
Pokud zvolíme možnost logování na SD kartu, je nutné zvolit interval logování. (Funguje to i s 1s, každopádně u Arduina Una nedoporučuju, spíše tak 10s.) Arduino tak například každých 10s uloží data ze senzoru do souboru SDATA.CSV. Při resetu anebo při dosažení určitého počtu záznamů je soubor přejmenován na OLDXXXXX.CSV a zapisuje se nanečisto. Data v souboru jsou ve formátu:

UNIXTIME,TEMP,HUM. Každý záznam je na vlastním řádku. To by mělo relativně usnadnit práci s daty.



## MQTT+Home Assistent

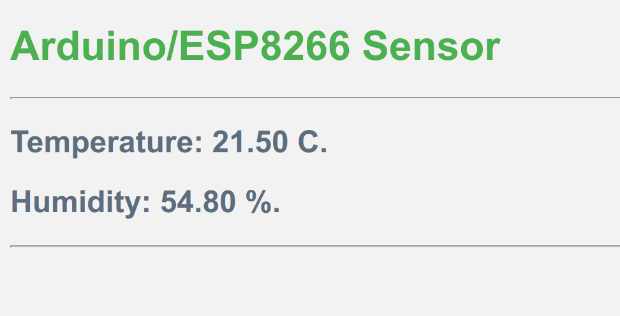
Tahle verze je trochu zajímavější, alespoň podle mého názoru. Pro testování jsem si stáhl Home Assistant OS, který obsahuje všechno potřebné ve verzi VM pro VirtualBox. Po nějaké základní konfiguraci jsem přidal addon na MQTT. Tento MQTT server se poté zadá do našeho kódu a Arduino umí odesílat data na MQTT, kde si je HomeAssistant převezme. S jednoduchou konfigurací v YAML můžeme přidat náš senzor (viz obrázek). V podstatě jen definujeme topicy, kam bude sensor posílat data. Ty musíme také samozřejmě definovat v našem kódu. Výsledkem je funkční entita v Home Assistentovi, kterou můžeme použít například na další automatizace.  


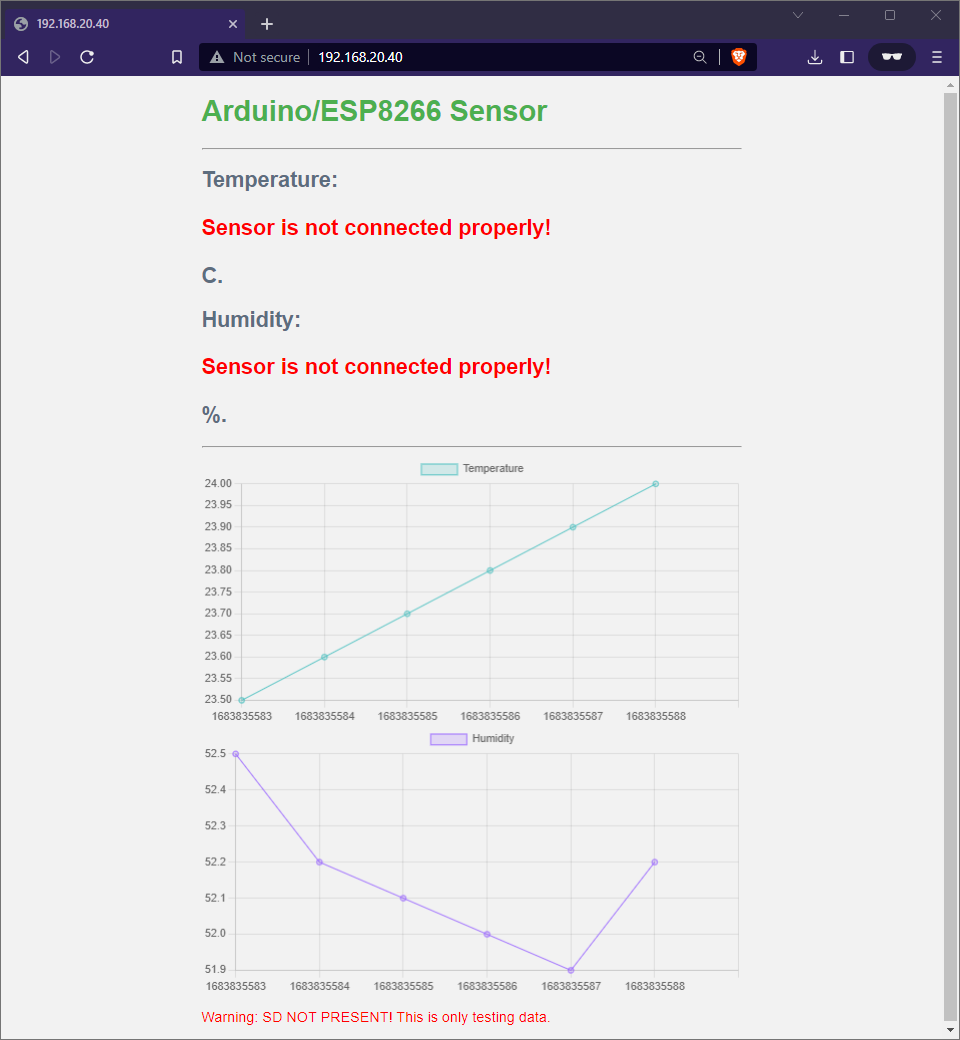


## ESP8266

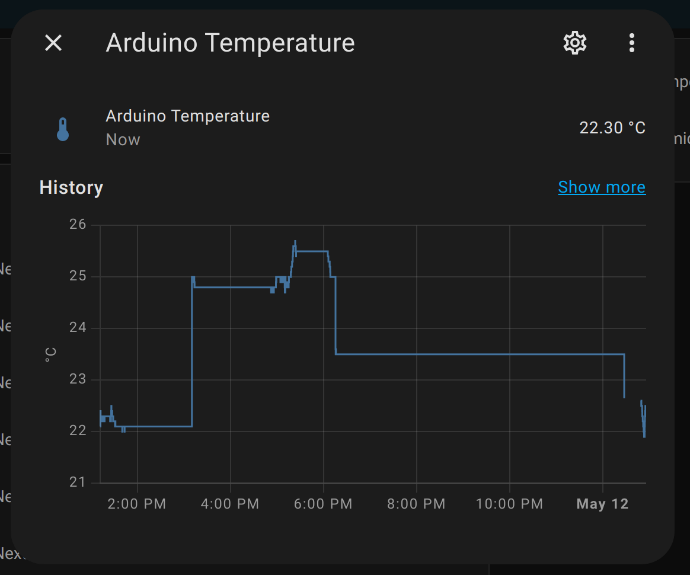
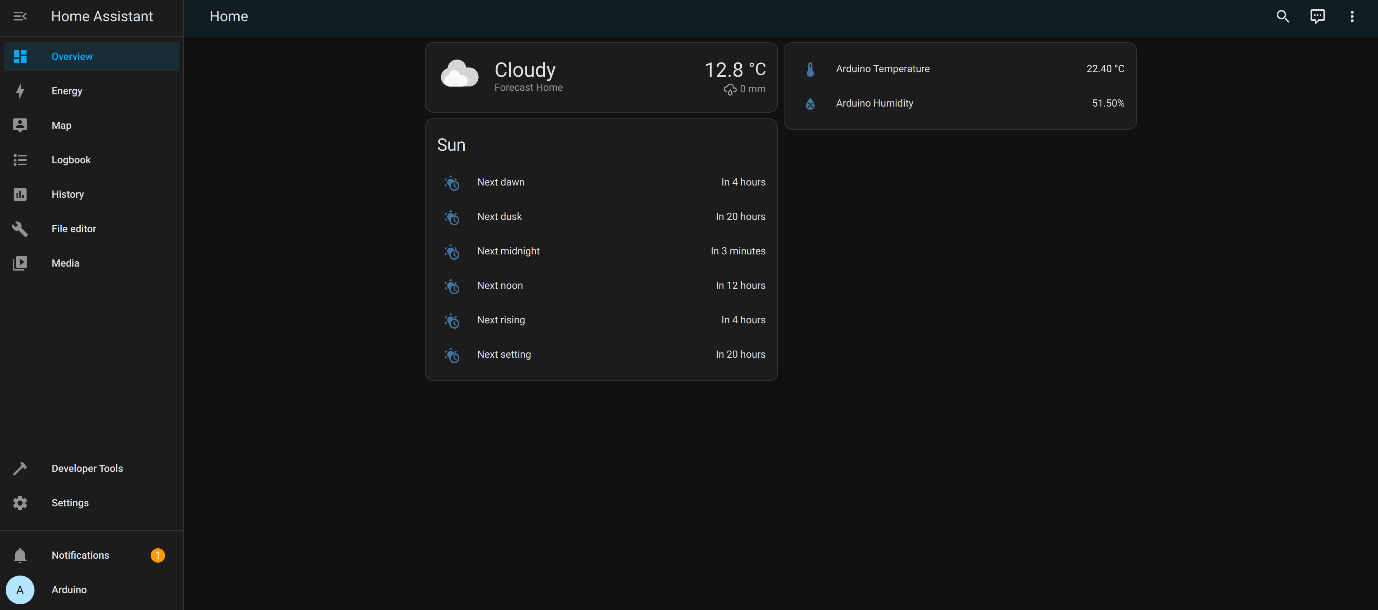
Protože mě to zajímalo a myslím, že to je celkem zajímavé i v kontextu této semestrální práce, upravil jsem ještě můj kód tak, aby mohl fungovat s ESP8266 jako hlavním procesorem místo Arduina Una. Tady jsem samozřejmě mohl použít knihovny na logování na SD kartu zároveň s odesíláním dat na MQTT, místo problém nebylo. Dokonce mi to i umožnilo udělat o něco hezčí UI pro mojí HTML stránku.

Problémem ale bylo, že můj shield kombinující Ethernet a SD kartu, který jsem doposud používal s Arduinem, není kompatibilní s mojí ESP deskou kvůli rozdílnému napětí. (Asi? Nejsem si upřímně moc jistý, ale tohle je nepravděpodobnější verze. Každopádně jsem to nikdy nezprovoznil.) Na to by byly potřeba další součástky, které jsem ale v době dělání hlavní částí práce neměl a díky ne příliš rychlému doručení lokálních e-shopů jsem je ani nestihl koupit a vyzkoušet. Každopádně jsem otestoval moje ESP minimálně s MQTT a senzorem a všechno fungovalo skvěle. Na webserver jsem i přidal http přihlášení a použil jsem tam nějakou JS knihovnu na vytvoření grafů z dat na SD kartě.





## Ukázka čidla v Home Assistantovi

(Graf není pěkný kvůli tomu, že jsem to nenechal běžet dost dlouho a často jsem to restartoval…)

## Závěr

Myslím, že se mi povedlo vytvořit relativně funkční verzi DIY domácího teplotního a vlhkostního senzoru. Pokud bych měl 3D tiskárnu, tak by byl dobrý nápad třeba vytisknout nějakou krabičku a udělat z toho trochu více permanentní řešení. Po tom, co jsem to mohl vyzkoušet, silně doporučuju na podobné projekty používat ESP8266/ESP32 a pokud bych to dělal od začátku taky, narazil bych na daleko méně problémů. Je lepší projít si tímhle vším než koupit nějaké smart-home teplotní čidlo? Spíše ne, ale i tak mě tento projekt bavil a kód co jsem vytvořil umožňuje relativně dobře porovnat vlastnosti Arduina a ESP.

# HW dokumentace

Zapojení jsem už popsal v teoretické případě a nic se v něm nemění. Závisí, jaký způsob ukládání dat zvolíte a podle toho je potřeba mít součástky. Je teoreticky možné použít více senzorů než jeden, ale kód by musel být upraven, aby s tím dokázal pracovat správně.

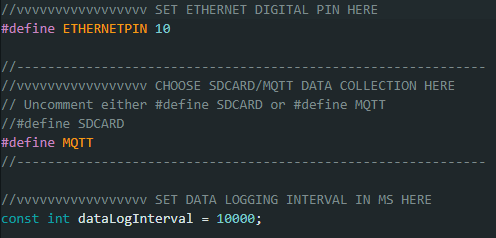
Pro mě to bylo primárně: (verze Arduino)

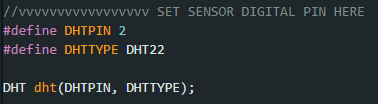
* PIN2 – teplotní čidlo DHT22
* PIN4 – SPI CS pro čtečku SD karet
* PIN10 – W5100 Ethernet

# Manuál

Odkaz na můj Gitlab repozitář, kde najdete všechen kód: <https://gitlab.fit.cvut.cz/svoboo30/ard-semestralka> případně mirror na githubu (pokud nemáte práva): <https://github.com/ondra2305/ard-tempsensor>.

Jak už bylo zmínění v kódu je potřeba definovat několik věcí, včetně zvoleného způsobu ukládání dat a použitých pinů. Mimo toho také (podle použití) MAC adresu pro Ethernet, MQTT server a topicy, NTP Server a časovou zónu, frekvenci logování dat v ms.



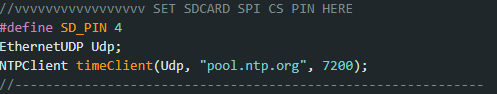


MQTT nebo SD kartu vybereme odkomentováním jednoho a zakomentováním toho co nepoužijeme. V kódu jsou podmíněné bloky, které ho umožňují zkompilovat podle naší volby. Většinu chyb jsem se pokusil, pokud možno přímočaře okomentovat v debug zprávách, co by se vám měly zobrazit v sériové konzoli při připojení k PC.

## SD karta

Pokud používáme SD kartu, je nutné ji nejdříve zformátovat ve formátu FAT16/FAT32. Je taky potřeba aby karta byla maximálně typu SDHC, tedy karty větší, jak 32 GB nebudou fungovat. Potom musíme nastavit pin, kde je SPI rozhraní (chip select) SD karty připojeno. Dále je potřeba nastavit NTP server (například pool.ntp.org nebo time.nist.gov) a offset od naší časové zóny oproti GMT (v zimě 1 h, v létě 2 h => 3600/7200).

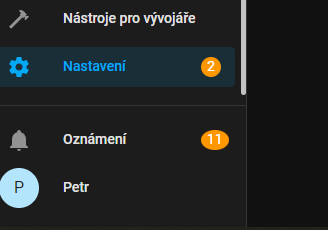
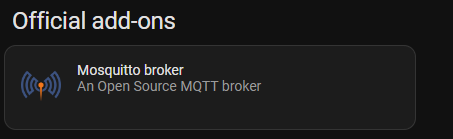
Pokud všechno funguje, stačí navštívit adresu Arduina (pokud nevíte, najdete jít v debug logu seriové konzole po spuštění programu), například <http://192.168.1.10> a měli bychom vidět naší aktuální teplotu a vlhkost! Na vaší SD kartu by se měla ukládat všechna data ve formátu CSV, rozdělená po jednotlivých souborech v případě resetu zařízení.

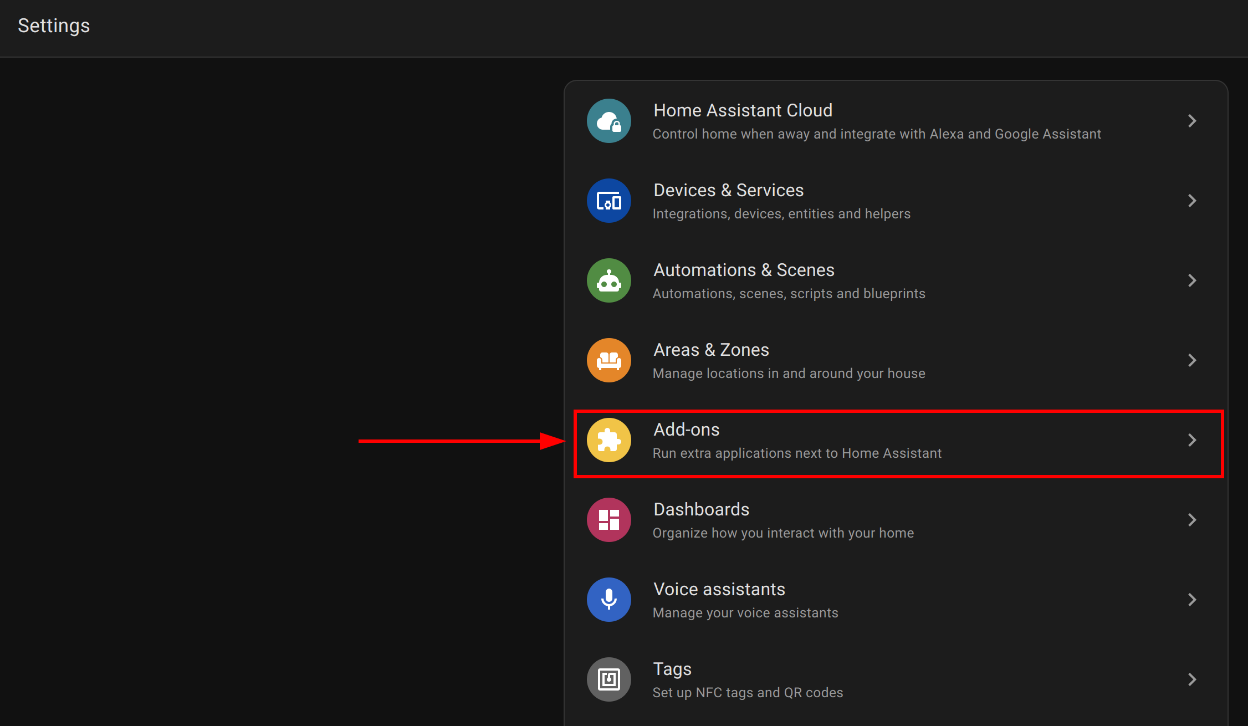


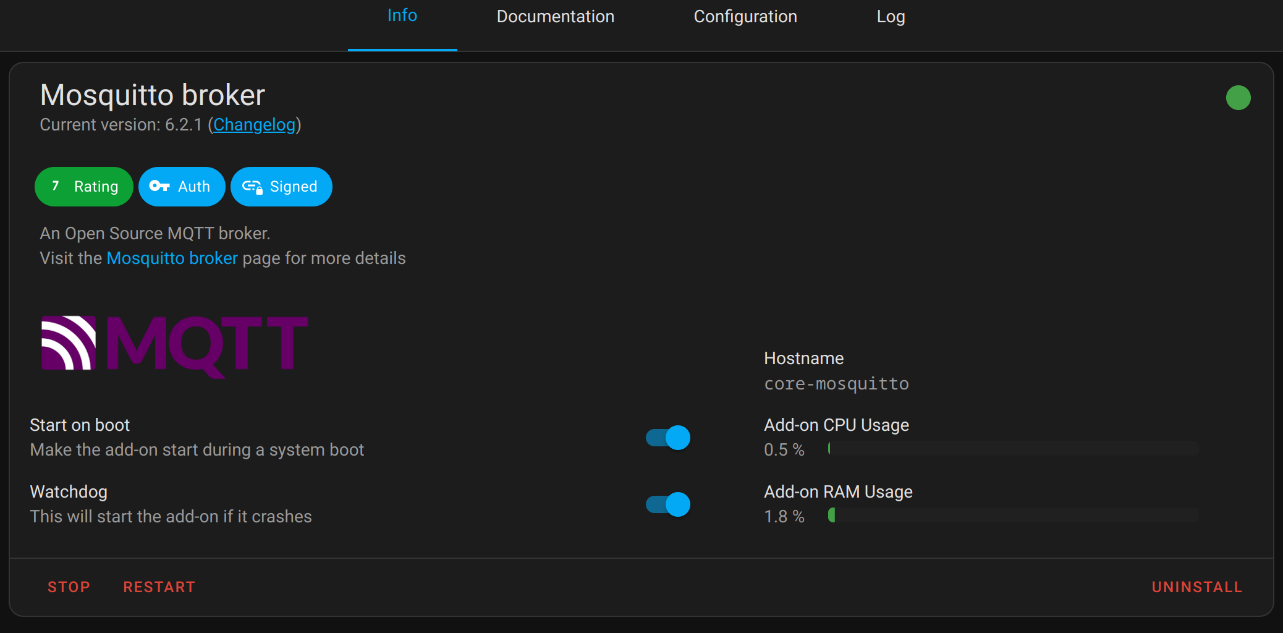
## MQTT & Home Assistant

Zde je konfigurace trochu složitější, ale pokusím se to shrnout co nejjednodušeji. Pro instalaci HomeAssistenta vás musím odkázat na jeho dokumentaci, liší se podle zařízení: <https://www.home-assistant.io/installation/>. Ale já doporučuju Home Assistant OS na Raspberry Pi nebo nějakém vašem počítači. Po spuštění a instalaci projdete prvním setup, kde je potřeba vytvořit účet. Uživatelské jméno a heslo si zapamatujte, bude potřeba pro připojení k MQTT.

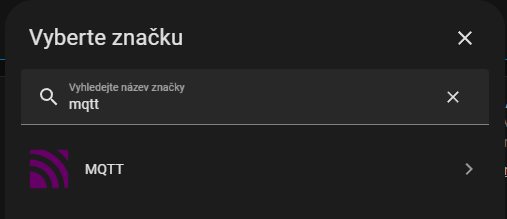
Dalším krokem je přidání MQTT addonu a jeho nastavení. V levé části obrazovky klikneme na nastavení, a potom Doplňky (Addons). Potom klikneme na obchod s doplňky a hned mezi prvními by mělo být „Mosquitto broker“, na to klikneme a nainstalujeme.

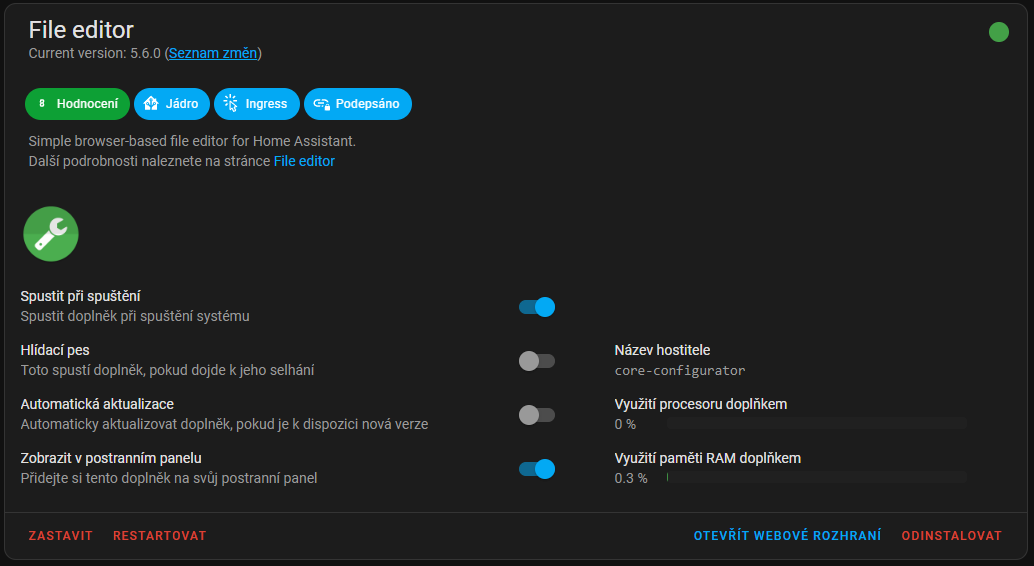


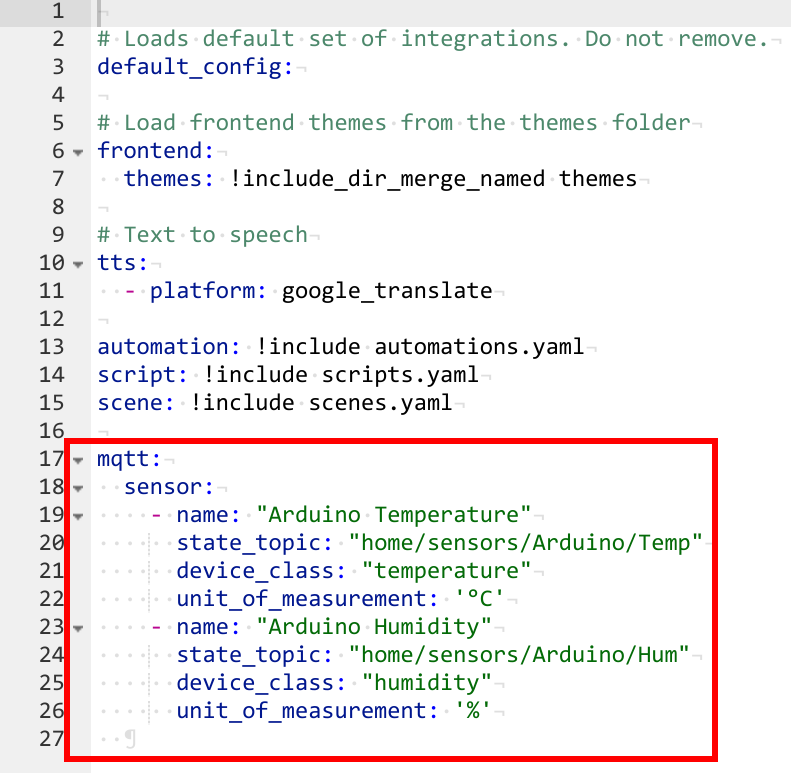


Potom je potřeba doplňek spustit. Dalé jděte do Nastavení – Zařízení a služby – Přidat integraci a vyberte MQTT. Tady by mělo stačit proklikat výchozí nastavení.



A to je vše pro MQTT! Posledním krokem je upravení konfigurace v YAML. Můžete otevřít konfigurační soubor configuration.yaml různými způsoby, ale asi nejlepší možnost je další doplněk – File editor. Ten potom stačí otevřít, najít správný soubor a potom už je upravit několik řádků konfigurace.

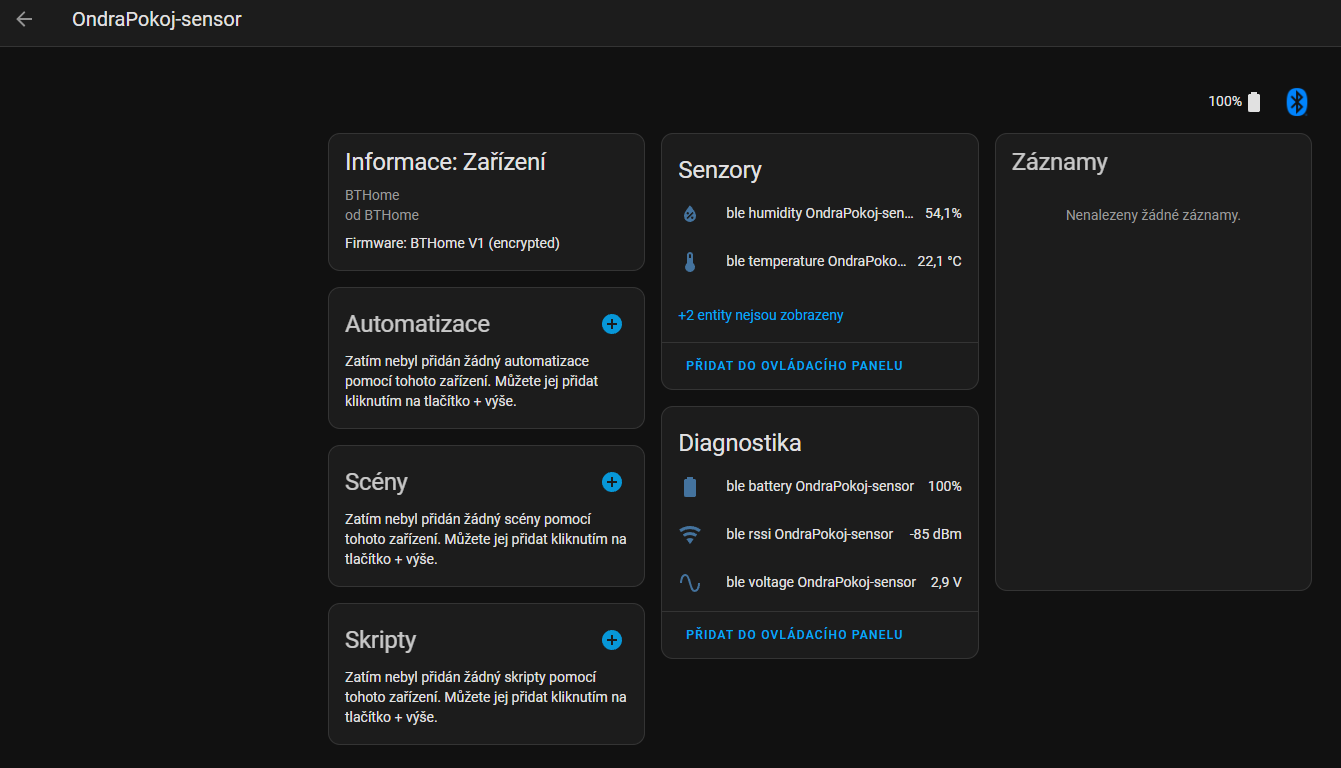


Pro základní konfiguraci můžete použít tento můj kód: 

Posledním krokem je přidání topiců, které jsme si výše definovali do našeho kódu. To uděláme ve vyznačené sekci na obrázku níže. Zde je potřeba použít stejné jako v HA. IP adresa bude IP adresa našeho serveru a username a password bude stejný jako váš účet v Home Assistantovi.



Pokud se vám to povedlo, měla by komunikace nyní probíhat. Také byste měli v Nastavení – Zařízení a služby – Zařízení (úplně nahoře) vidět váš sensor. Když ho vyberete měli byste se dostat na obrazovku kde budou všechny jeho data (viz obrázek níže). Pokud chcete, můžete si ho rovnou přidat do ovládacího panelu tlačítkem „Přidat do ovládacího panelu“.



(Toto je příklad jiného sensoru, rozhraní ale vypadá přibližně stejně.)

## ESP8266 verze

Rozdíly ve verzi pro ESP8266 už nebudu v návodu rozepisovat, největší je asi zadání názvu wifi sítě a hesla podobně jako jsme zadávali například piny v kódu. Mělo by být možné používat jak SD kartu, tak MQTT, ale neměl jsem možnost to podrobně otestovat.