|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Teploměr** | | |
| Matěj Bezděk | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2017/2018 | |

#### Poděkování

Děkuji panu učiteli Ing. Petru Grussmanovi za pomoc při výběru správného hardwaru a následné konzultaci.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2017

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvoření funkčního zařízení, které bude měřit teplotu a následně ji odesílat do online tabulkového editoru Google Sheets. Ke komunikaci mezi vývojovou deskou a Google Sheets dochází skrze Wi-Fi připojení. Teploměr byl vyvinut ve vývojovém prostředí Arduino.IDE a využívá také Google Apps Scripty.

**Klíčová slova:** Teploměr; Arduino; DS18B20; ESP32;

OBSAH

[Úvod 4](#_Toc502503396)

[1 Teoretická a metodická východiska 5](#_Toc502503397)

[2 Využité technologie 6](#_Toc502503398)

[2.1 Vývojová deska 6](#_Toc502503399)

[2.2 Teplotní senzor 7](#_Toc502503400)

[2.3 Použitý software 7](#_Toc502503401)

[2.3.1 Arduino 1.8.5 7](#_Toc502503402)

[2.3.2 Google Sheets 7](#_Toc502503403)

[2.3.3 Google Apps Script 7](#_Toc502503404)

[3 Způsoby řešení a použité postupy 8](#_Toc502503405)

[3.1 Seznam nutných součástek 8](#_Toc502503406)

[3.2 Hardware 8](#_Toc502503407)

[3.3 Základní funkce jazyka Arduino 9](#_Toc502503408)

[3.4 Google Sheets 9](#_Toc502503409)

[3.5 Google Apps Script 10](#_Toc502503410)

[3.6 HTTPSRedirect 10](#_Toc502503411)

[4 Výsledky řešení, výstupy, uživatelský manuál 11](#_Toc502503412)

[4.1 Připojení k Wi-Fi 11](#_Toc502503413)

[4.2 Deep Sleep 11](#_Toc502503414)

[Závěr 13](#_Toc502503415)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 14](#_Toc502503416)

Úvod

Dlouho jsem uvažoval, jaký závěrečný projekt mě bude bavit sestavit a neskončí po prezentaci v šuplíku. Myslím si, že pro mnoho lidí je teploměr opravdu důležitá věc k životu. Nedílnou součástí je i v naší domácnosti a jelikož si moje babička již několik let vypisuje každodenní naměřené hodnoty, tak mě napadlo, že by se tohle dalo zautomatizovat.

Cílem tohoto projektu bylo vytvoření funkčního zařízení, které bude měřit teplotu a zapisovat hodnotu do online textového editoru.

První část mé dokumentace se zaměřuje na principy a postupy, které jsem využil k začátku snadného měření. V druhé části popisuji postup zapisování hodnot na server.

# Teoretická a metodická východiska

Na začátku projektu bylo důležité vybrat si správné technologie, které budou použity. Na internetu jsem hledal správnou vývojovou desku, která bude vše řídit. Po domluvě s panem učitel jsem použil nástupce čipu EPS8266, čip ESP32 s integrovaným Wi-Fi modulem, který je na projekt potřeba.

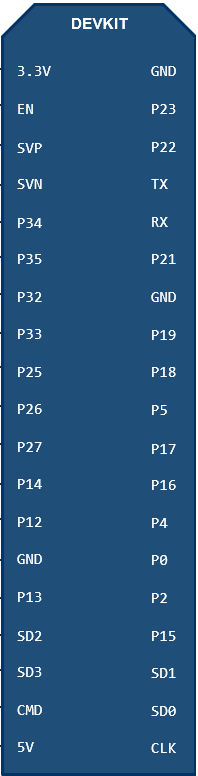
Dále bylo potřeba zvolení teplotního senzoru DS18B20 a správný rezistor. Na internetu jsem si našel správné zapojení teplotního senzoru a následně jsem vše propojil na nepájivém poli.

# Využité technologie

## Vývojová deska

ESP32 Development board je vývojová deska pro Arduino. Tato deska obsahuje nejnovější čip ESP32 od firmy Espressif, který má oproti předchozímu [ESP8266](http://navody.arduino-shop.cz/navody-k-produktum/esp8266-vyvojova-deska-wemos-d1.html) nejen větší výkon, ale i nové funkce. Oproti staršímu čipu obsahuje nově ESP32 vedle Wi-Fi také Bluetooth ve verzi 4.2 s podporou BLE (Bluetooth Low Energy).

Parametry:

* Napájení 3.3V
* Flash paměť 4MB
* SRAM 520 kB
* Wi-Fi: [802.11](https://en.wikipedia.org/wiki/IEEE_802.11) b/g/n
* 36 GPIO pinů
* Dvoujádrový procesor s frekvencí 160 MHz

*Obrázek č1. Datasheet ESP32*

## Teplotní senzor

Na měření teploty jsem použil teplotní senzor DS18B20 od firmy Maxim. Umožňuje měřit v rozsahu -10 až +85 stupňů Celsia s garantovanou přesností ±0,5 °C. Pro komunikaci s ESP deskou jsou využité knihovny OneWire a DallasTemperature.

## Použitý software

### Arduino 1.8.5

Arduino IDE je Open-Source platforma pro snadný návrh a vývoj elektronických programovatelných zařízení. Program obsahuje možnost kompilovat kód přímo na desku, má intuitivní ovládání a díky široké komunitě příznivců Arduino se případné problémy řeší snadno.

### Google Sheets

Google Sheets je online tabulkový editor, který vás nechá vytvořit a formátovat tabulky a pracovat s ostatními lidmi. S Google Sheets si můžete vytvořit a editovat tabulky přímo ve vašem prohlížeči - není potřeba žádný software. Několik lidí může pracovat zároveň, můžete sledovat změny, které se automaticky ukládají.

### Google Apps Script

Google Apps Script je skriptovací jazyk pro vývoj aplikací v Google Apps platformě. Je založen na JavaScriptu 1.6 s přídělem z 1.7 a 1.8. Google Apps Script poskytuje lehkou cestu k automatizaci Google produktů. Apps Script je také nástroj, který umožňuje doplňky pro dokumenty Google Dosc, Sheets and Slides.

# Způsoby řešení a použité postupy

## Seznam nutných součástek

* Vývojová deska ESP32,
* teplotní senzor DS18B20,
* rezistor 4,7kΩ,
* nepájivé pole,
* vodiče.

## Hardware

Prvním krokem bylo správné propojení součástek a následné nastavení vývojové desky ESP32 ve vývojovém prostředí Arduino.IDE. Na internetu jsem si stáhnul potřebné knihovny, které byly nutné ke komunikaci. Zde nastal první problém, ESP32 vyšlo teprve nedávno a často jsem na internetu našel nefunkční knihovny.Po zdlouhavém hledání a častých chybách při kompilaci jsem našel fungující knihovnu. Dále bylo potřeba stáhnout knihovny OneWire a DallasTemperature ke komunikaci s teplotním senzorem, který jsem si připojil na pin 4.

// připojení knihoven

#include <OneWire.h>

#include <DallasTemperature.h>

// nastavení čísla vstupního pinu

const int pinCidlaDS = 4;

OneWire oneWireDS(pinCidlaDS);

DallasTemperature senzoryDS(&oneWireDS);

## Základní funkce jazyka Arduino

Jazyk Arduino obsahuje dvě základní funkce. Funkce setup() a loop(). Funkce setup() slouží k inicializaci proměnných a nastavení potřebných hodnot. Je zavolána při spuštění nebo po restartu. Funkce loop() je hlavní funkce programu, která se stále opakuje.

Ve funkci loop() jsem si vypsal hodnoty a měření fungovalo.

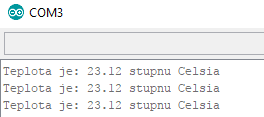
void loop(void) {

senzoryDS.requestTemperatures();

Serial.print("Teplota je: ");

Serial.print(senzoryDS.getTempCByIndex(0));

Serial.println(" stupnu Celsia");

 delay(1000);

}

*Obrázek č2. Základní měření*

## Google Sheets

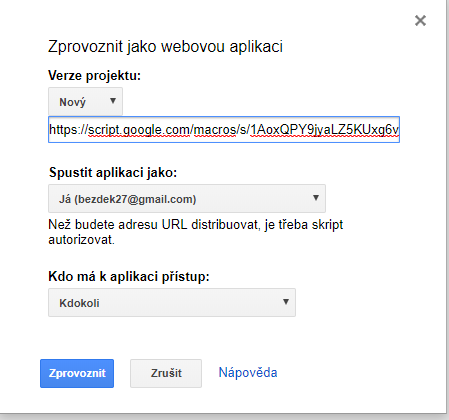
Základní měření funguje, následně si potřebuji vytvořit online tabulkový editor, kde se hodnoty budou zapisovat. Po registrování se a vytvoření své tabulky si zkopíruju potřebné GoogleSheetsID, které najdu v URL. Je to unikátní kód, který budu později k propojení Google Apps Scriptu a GoogleSheets. ESP32 má potřebný Wi-Fi modul, takže připojení k internetu není problém.

## Google Apps Script

## <TOTO Rozepsat postup popis jazyka …. jak se dostanete do prostredi skriptu debug skriptu ...

Po vytvoření editoru je důležité vytvořit script, který dokáže právě s editorem správně komunikovat. Google Apps Script byla pro mě novinka, ale naštěstí jsem si na internetu našel potřebný návod.

Použil jsem funkce doGet{} a funkci saveData{}, kde se pomocí GoogleSheetsId připojily potřebné data. Následně bylo potřeba skript uložit a publikovat ho jakou veřejnou stránku.



*Obrázek č3. Editování stránky*

## HTTPSRedirect

ESP32 potřebuje správně dekódovat informace z hlavičky získané z prvního serveru, aby mohl extrahovat přesměrovanou URL a mohl poslat druhý GET požadavek na nový server. K tomuhle jsem použil vytvořenou knihovnu HTTPSRedirect, kterou jsem si stáhnul z Githubu. Tenhle způsob se dá použít i jinou možností, například Pushingboxem ( rozepsat), ale ten mi nefungoval správně.

# Výsledky řešení, výstupy, uživatelský manuál

## Připojení k Wi-Fi

Vše potřebné mám již připravené. Nyní stačí naprogramovat kód ve vývojovém prostředí Arduino IDE. Začínám připojením potřebných knihoven a připojením na Wi-Fi.

<https://github.com/tzapu/WiFiManager>

predelat pokud možno na tutuo knihovnu

//připojení potřebných knihoven.

#include "HTTPSRedirect.h"

#include <OneWire.h>

#include <DallasTemperature.h>

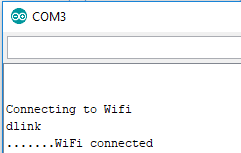
#define ONE\_WIRE\_BUS 4 //Port číslo 4

OneWire oneWire(ONE\_WIRE\_BUS); //

DallasTemperature sensors(&oneWire); //

const char\* ssid = "dlink"; //připojení k wi-fi síti

const char\* password = "29126170";



*Obrázek č4. Připojení k Wi-Fi*

## Deep Sleep

Ke svému projektu jsem použil také funkci na uspání mé vývojové desky. Na internetu jsem si našel tři druhy funkcí. Modem sleep, light sleep a také funkci deep Sleep. Nakonec jsem se rozhodl použít funkci Deep sleep. Tato funkce dokáže na nastavený čas „uspat“ mojí vývojovou desku a tímhle ušetřit spotřebu elektrické energie. Tímhle jsem docílil, aby se mi nezapisovala hodnota pořád, ale například co každou hodinu. Aby fungovala tato funkce správně, stačí propojit vodiči dva piny, RST a D0 a následně ji správně vyvolat ve vývojovém prostředí.

//Nastavení Deep Sleep na 15 minut

Const int deepSleep = 900000000;

//Následně ve funkci void() dojde ke spánku

ESP.deepSleep(deepSleep);

# Závěr

Cílem mého projektu bylo vytvoření teplotního čipu, který bude hodnoty zapisovat do Google Kalendáře, v kterém bude možno hodnoty vyhledávat a následně připisovat poznámky ke každému dni. Po zjištění, že Google Kalendář nepovoluje hodnoty získávat přímo z ESP, ale jen vypisovat hodnoty již zapsané v kalendáři, jsem musel vymyslet jiné řešení. Mým řešením bylo projekt zjednodušit a získané hodnoty z ESP vypisovat pouze do textového editoru Google Sheets. Dalším problémem bylo to, že ESP32 je docela nová vývojová deska a bylo těžké najít potřebné funkční knihovny.

V priloze vsechny kody

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] ESP8266 Deep Sleep [online]. Dostupné také z: https://randomnerdtutorials.com/esp8266-deep-sleep-with-arduino-ide/

[2] ESP32[online]. Dostupné také z: <http://navody.arduino-shop.cz/navody-k-produktum/vyvojova-deska-esp32.html>

[3] HTTPS[online]. Dostupné také z : htts://github.com/electronicsguy/ESP8266/tree/master/HTTPSRedirect

[4] Google Script[online]. Dostupné také z: https://www.google.com/script/start/