Instalace CMLSmartHome

Radek Pátek

14.04.2019

Obsah

[1. Úvod 3](#_Toc56963563)

[2. Architektura 3](#_Toc56963564)

[3. CMLSmartHomeController 3](#_Toc56963565)

[3.1. Raspbian 3](#_Toc56963566)

[3.2. Samba 3](#_Toc56963567)

[3.2.1. Úprava fstab 4](#_Toc56963568)

[3.3. Webový server 4](#_Toc56963569)

[3.4. Databázový server 5](#_Toc56963570)

[3.4.1. Nastavení hesla a zabezpečení 6](#_Toc56963571)

[3.4.2. Vytvoření databáze 6](#_Toc56963572)

[3.5. Service 8](#_Toc56963573)

[4. CMLSmartHomeCollector 8](#_Toc56963574)

[5. CMLSmartHomeWeb 9](#_Toc56963575)

[5.1. Service 9](#_Toc56963576)

[6. CMLSmartHomeDashboard 10](#_Toc56963577)

[6.1. Zdrojový kód 10](#_Toc56963578)

[6.2. Podkladová bitmapa 10](#_Toc56963579)

[6.3. Vytvoření písem 11](#_Toc56963580)

## Úvod

Dokument popisuje instalaci jednotlivých prvků systému CMLSmartHome.

## Architektura

## CMLSmartHomeController

CMLSmartController je instalován na Raspberry Pi 3 s operačním systémem Raspbian.

## Raspbian

Instalace dotnetcore runtime:

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install libunwind8

$ sudo apt-get install wiringpi

$ sudo wget <https://dotnetcli.blob.core.windows.net/dotnet/aspnetcore/Runtime/2.1.0/aspnetcore-runtime-2.1.0-linux-arm.tar.gz>

$ sudo tar -xvf aspnetcore-runtime-2.1.0-linux-arm.tar.gz -C /opt/dotnet/

$ sudo ln -s /opt/dotnet/dotnet /usr/local/bin

$ sudo dotnet --info

Instalace grafické knihovny

$ sudo apt-get install –y libgdiplus

Vytvoření adresáře pro aplikaci:

$ sudo mkdir /var/aspnetcore

$ sudo mkdir /var/aspnetcore/CMLSmartHomeController

## Samba

Instalace:

$ sudo apt-get install samba samba-common-bin

$ sudo mkdir /mnt/smb

Úprava Hosts:

$ sudo nano /etc/hosts

do souboru se vloží:

192.168.1.103 DISKSTATION

Vytvoření souboru s credential:

$ sudo mkdir /etc/samba/credentials

$ sudo nano /etc/samba/credentials/DISKSTATION

do souboru se vloží:

username=Radek

password=koniklec

## Úprava fstab

$ sudo nano /etc/fstab

Do /etc/fstab se přidá řádek:

//DISKSTATION/Others/Develop /mnt/smb cifs credentials=/etc/samba/credentials/DISKSTATION 0 0

## Webový server

Instaluje se webový server Nginx

$ sudo apt-get install nginx

$ sudo service nginx start

Úprava souboru  /etc/nginx/sites-available/default

server {

listen 80;

location /api {

proxy\_pass http://localhost:5000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location / {

proxy\_pass http://localhost:5002;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

server {

listen 443 ssl;

listen 8443 ssl;

ssl on;

ssl\_certificate /etc/nginx/my\_ssl\_certs/cert.pem;

ssl\_certificate\_key /etc/nginx/my\_ssl\_certs/privkey.pem;

ssl\_session\_timeout 5m;

ssl\_protocols SSLv3 TLSv1 TLSv1.1 TLSv1.2;

server\_name patek.cloud;

location /api {

proxy\_pass http://localhost:5000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location / {

proxy\_pass http://localhost:5002;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

Vygenerovaní dhparam certifikatu

openssl dhparam -out dhparam.pem 4096

Uložení SSL certifikátů

sudo mkdir /etc/nginx/my\_ssl\_certs

sudo cp /mnt/smb/CMLSmartHome/Cert/privkey.pem /etc/nginx/my\_ssl\_certs

sudo cp /mnt/smb/CMLSmartHome/Cert/cert.pem /etc/nginx/my\_ssl\_certs

## Databázový server

CMLSmartHomeController používá pro persistenci dat databázové řešení MariaDB. Nejdříve se provede instalace DB serveru:

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install mariadb-server

## Nastavení hesla a zabezpečení

Dalším krokem je konfigurace databáze, kde je důležité nastavení hesla pro uživatele root.

$ sudo mysql\_secure\_installation

-----------

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current

password for the root user. If you've just installed MariaDB, and

you haven't set the root password yet, the password will be blank,

so you should just press enter here.

Enter current password for root (enter for none):

Následně se odpoví na následující otázky:

* Change the root password? [Y/n] – **No**
* Remove anonymous users? [Y/n] - **Yes**
* Disallow root login remotely? [Y/n] - **No**
* Remove test database and access to it? [Y/n] - **Yes**
* Reload privilege tables now? [Y/n] – **Yes**

Pro vzdálené připojení k db je nutné upravit atribut *bind-address* z hodnoty *127.0.0.1* na hodnotu *0.0.0.0*

$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

úprava:

bind-address = 0.0.0.0

Pro potřeby vzdálené administrace DB je potřeba povolit vzdálený přístup uživatele *root* z jiných stanic sítě než *localhost (127.0.0.1).*

$ mysql -u root –p

mysql> use mysql

mysql> GRANT ALL ON \*.\* to root@'192.168.1.%' IDENTIFIED BY 'borovnicka76';

mysql> SELECT user, host FROM user;

mysql> FLUSH PRIVILEGES;

Na závěr se restartuje služba DB serveru, aby se projevila změna konfigurace

sudo service mysql restart

## Vytvoření databáze

V nastaveném serveru se vytvoří DB *SmartHome*

-- --------------------------------------------------------

-- Hostitel: 192.168.1.152

-- Verze serveru: 10.1.23-MariaDB-9+deb9u1 - Raspbian 9.0

-- OS serveru: debian-linux-gnueabihf

-- HeidiSQL Verze: 9.4.0.5125

-- --------------------------------------------------------

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!50503 SET NAMES utf8mb4 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

-- Exportování struktury databáze pro

CREATE DATABASE IF NOT EXISTS `SmartHome` /\*!40100 DEFAULT CHARACTER SET cp1250 COLLATE cp1250\_czech\_cs \*/;

USE `SmartHome`;

-- Exportování struktury pro tabulka SmartHome.Collectors

CREATE TABLE IF NOT EXISTS `Collectors` (

`Id` bigint(20) NOT NULL AUTO\_INCREMENT,

`Name` longtext COLLATE cp1250\_czech\_cs,

`MACAddress` longtext COLLATE cp1250\_czech\_cs,

`Description` longtext COLLATE cp1250\_czech\_cs,

`SmartHomeControllerId` bigint(20) DEFAULT NULL,

PRIMARY KEY (`Id`),

KEY `IX\_Collectors\_SmartHomeControllerId` (`SmartHomeControllerId`),

CONSTRAINT `FK\_Collectors\_Controller\_SmartHomeControllerId` FOREIGN KEY (`SmartHomeControllerId`) REFERENCES `controller` (`Id`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=254 DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

-- Exportování struktury pro tabulka SmartHome.Controller

CREATE TABLE IF NOT EXISTS `Controller` (

`Id` bigint(20) NOT NULL AUTO\_INCREMENT,

`Name` longtext COLLATE cp1250\_czech\_cs,

`MACAddress` longtext COLLATE cp1250\_czech\_cs,

`Description` longtext COLLATE cp1250\_czech\_cs,

PRIMARY KEY (`Id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

-- Exportování struktury pro tabulka SmartHome.Dashboard

CREATE TABLE IF NOT EXISTS `Dashboard` (

`Id` bigint(20) NOT NULL AUTO\_INCREMENT,

`OutdoorCollectorId` bigint(20) DEFAULT NULL,

PRIMARY KEY (`Id`),

KEY `IX\_Dashboard\_OutdoorCollectorId` (`OutdoorCollectorId`),

CONSTRAINT `FK\_Dashboard\_Collectors\_InternalCollectorId` FOREIGN KEY (`InternalCollectorId`) REFERENCES `Collectors` (`Id`) ON DELETE NO ACTION,

CONSTRAINT `FK\_Dashboard\_Collectors\_OutdoorCollectorId` FOREIGN KEY (`OutdoorCollectorId`) REFERENCES `Collectors` (`Id`) ON DELETE NO ACTION

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

-- Exportování struktury pro tabulka SmartHome.SensorRecords

CREATE TABLE IF NOT EXISTS `SensorRecords` (

`Id` bigint(20) NOT NULL AUTO\_INCREMENT,

`SensorId` bigint(20) NOT NULL,

`CollectorId` bigint(20) NOT NULL,

`Value` double NOT NULL,

`Unit` int(11) NOT NULL,

`DateTime` datetime(6) NOT NULL DEFAULT '0001-01-01 00:00:00.000000',

PRIMARY KEY (`Id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2456 DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

-- Exportování struktury pro tabulka SmartHome.Sensors

CREATE TABLE IF NOT EXISTS `Sensors` (

`Id` bigint(20) NOT NULL AUTO\_INCREMENT,

`Name` longtext COLLATE cp1250\_czech\_cs,

`Type` int(11) NOT NULL,

`CollectorId` bigint(20) DEFAULT NULL,

`Unit` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`Id`),

KEY `IX\_Sensors\_CollectorId` (`CollectorId`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

-- Exportování struktury pro tabulka SmartHome.\_\_EFMigrationsHistory

CREATE TABLE IF NOT EXISTS `\_\_EFMigrationsHistory` (

`MigrationId` varchar(95) COLLATE cp1250\_czech\_cs NOT NULL,

`ProductVersion` varchar(32) COLLATE cp1250\_czech\_cs NOT NULL,

PRIMARY KEY (`MigrationId`)

) ENGINE=InnoDB DEFAULT CHARSET=cp1250 COLLATE=cp1250\_czech\_cs;

-- Export dat nebyl vybrán.

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

## Service

Vytvoří se služba pro CMLSmartHomeController:

$ sudo nano /etc/systemd/system/CMLSmartHomeController.service

do souboru se vloží:

[Unit]

Description=CML Smart Home

[Service]

WorkingDirectory=/var/aspnetcore/CMLSmartHomeController/

ExecStart=/opt/dotnet/dotnet /var/aspnetcore/CMLSmartHomeController/CMLSmartHomeController.dll

Restart=always

# Restart service after 10 seconds if the dotnet service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=CMLSmartHome

User=root

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false

[Install]

WantedBy=multi-user.target

Vytvoří se skript pro automatické nasazení služby:

$ sudo nano /var/aspnetcore/CMLSmartHomeControllerDeploy.sh

do souboru se vloží:

#!/bin/bash

sudo service nginx stop

sudo systemctl stop CMLSmartHomeController.service

sudo rm -r /var/aspnetcore/CMLSmartHomeController/\*

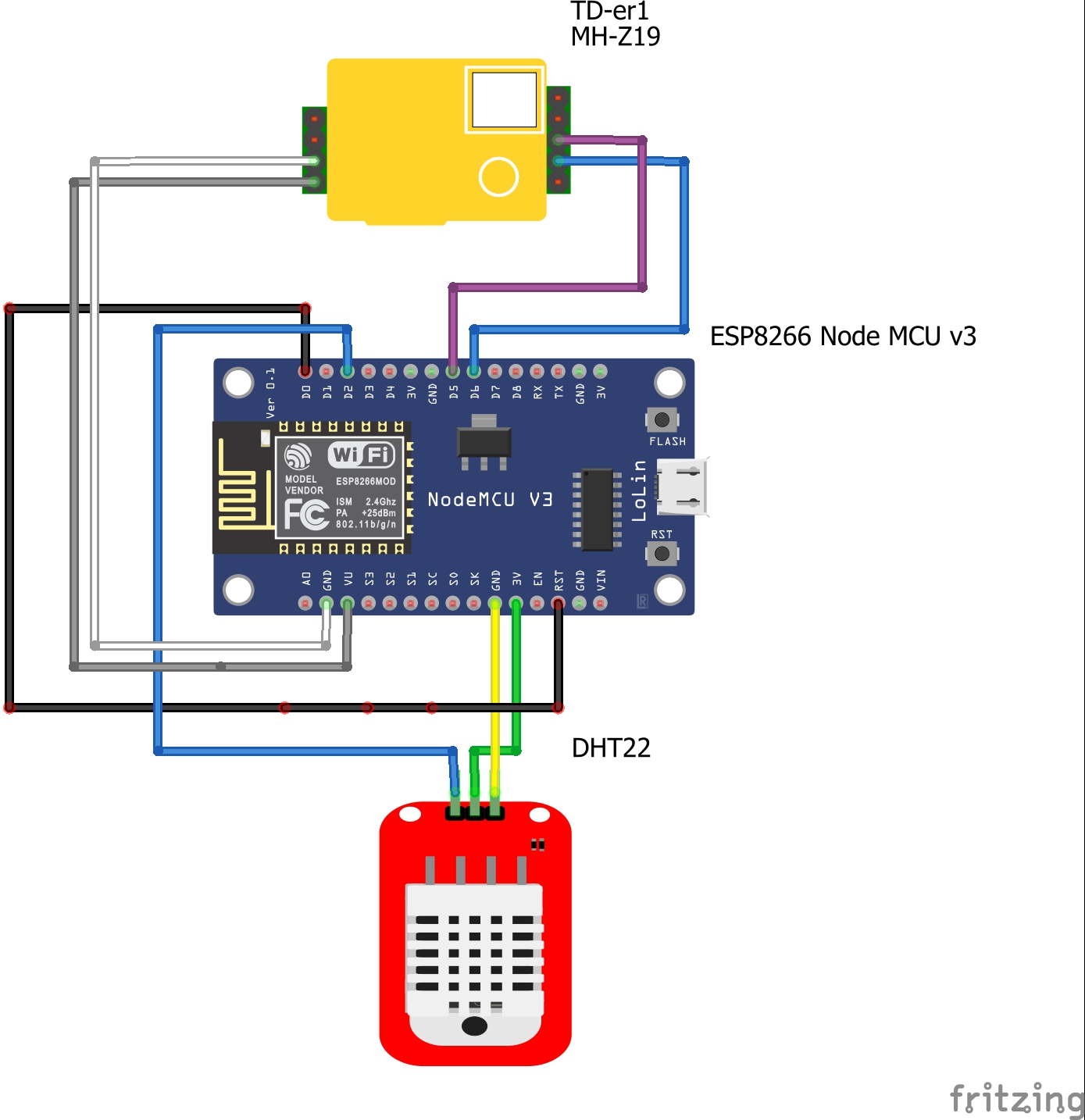
sudo cp -avr /mnt/smb/CMLSmartHome/CMLSmartHomeController/\* /var/aspnetcore/CMLSmartHomeController

sudo systemctl start CMLSmartHomeController.service

sudo service nginx start

$ sudo chmod +x /var/aspnetcore/CMLSmartHomeControllerDeploy.sh

## CMLSmartHomeCollector



## CMLSmartHomeWeb

Vytvoření adresáře pro aplikaci:

$ sudo mkdir /var/aspnetcore/CMLSmartHomeWeb

## Service

Vytvoří se služba pro CMLSmartHomeWeb:

$ sudo nano /etc/systemd/system/CMLSmartHomeWeb.service

do souboru se vloží:

[Unit]

Description=CML Smart Home Web

[Service]

WorkingDirectory=/var/aspnetcore/CMLSmartHomeWeb/

ExecStart=/opt/dotnet/dotnet /var/aspnetcore/CMLSmartHomeWeb/CMLSmartHomeWeb.dll

Restart=always

# Restart service after 10 seconds if the dotnet service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=CMLSmartHomeWeb

User=root

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false

[Install]

WantedBy=multi-user.target

## CMLSmartHomeDashboard

## Zdrojový kód

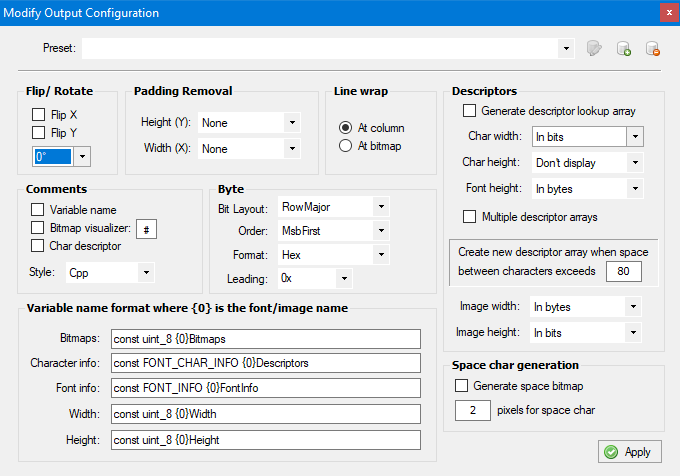
Zdrojový kód je spravován v GiHub (<https://github.com/radekpatek/CMLSmartHome>) . K editaci zdrojového kódu se používá Visual Studio Code.

## 

## Podkladová bitmapa

Pro vytvoření podkladové bitmapy se využije následující postup:

* V projektu GitHub ve složce /[CMLSmartHomeDashboard](https://github.com/radekpatek/CMLSmartHome/tree/master/CMLSmartHomeDashboard" \o "CMLSmartHomeDashboard)/src se edituje bitmapa Board.bmp
* V nástroji TheDotFactory [[1]](#footnote-1)(GitHub <https://github.com/pavius/the-dot-factory>) se provede transformace bitmapy do definice pole byte v haxa zápisu dle nastavení



* Vygenerovaný definice pole byte se přenese do /[CMLSmartHomeDashboard](https://github.com/radekpatek/CMLSmartHome/tree/master/CMLSmartHomeDashboard" \o "CMLSmartHomeDashboard)/src/Bitmap.h do konstanty dashboardImage

## Vytvoření písem

K vytvoření použitelných sad písem se použije také aplikace TheDotFactory

1. Aplilaci, která je stažena GitHub (<https://github.com/pavius/the-dot-factory> ) je klonovaná do lokálního adresáře C:\Users\radek\source\repos\the-dot-factory\TheDotFactory\. Aplikace vyžaduje .NET Framework 3.5 Service pack 1 [↑](#footnote-ref-1)