|  |
| --- |
| Mittermayr |
| Low-budget house heating system using Raspberry Pi |
| Memorandum |
|  |
| **Michael Mittermayr** |
| **7/6/2015** |

|  |
| --- |
| Unlocking anticipated technology power for tremendous energy efficiency gains by applying modern microcontroller technology combined with sophisticated algorithms to function as a low-budget housing heating system. |

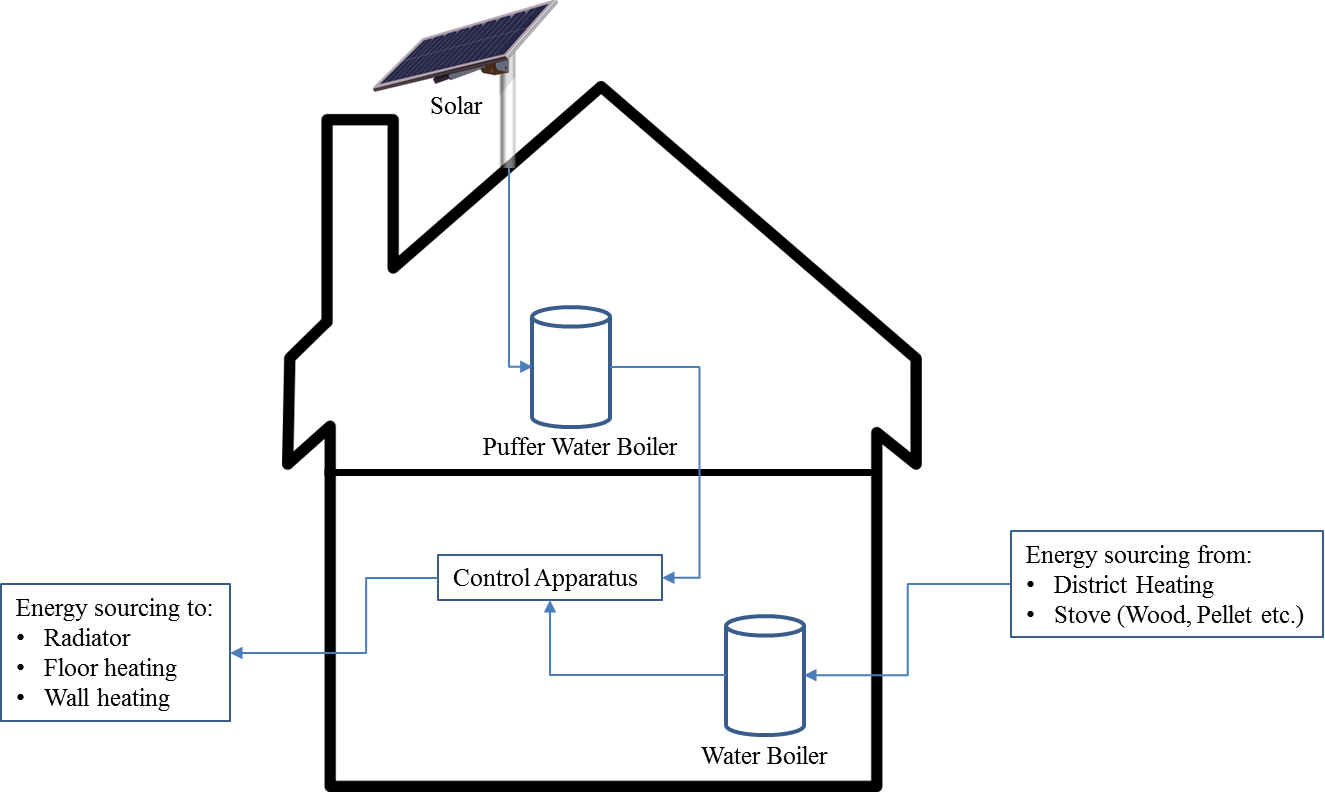
Contents

**No table of contents entries found.**

**No table of figures entries found.**

# Introduction

# Site Map



# Technical Site Map

# Heating circuit

A house consists of one or more heating circuits. Primarily purpose of a heating circuit is to give off heat by holding hot water. Each heating circuit comes with a pump that guarantees a smooth flow of warm water within this circuit. Equally important is the water mixer which regulates the amount of warmed water determined by one or more feedback temperature sensor. However, each circuit is characterised by different attitudes and procedures and can be applied as a radiator, wall heating, living room heating, floor heating and so on.

In our case we use:

* One heating circuit for radiators. (Heating circuit 1)
* One heating circuit for wall heating. (Heating circuit 2)
* One heating circuit for floor heating. (Heating circuit 3)
* One heating circuit for warm water treatment. (Heating circuit 4)

# Heating circuit map

Time control

**Heating Circuit 4**

(Warm Water Treatment)

* Pump 4
* Valve warm water district heating
* Sensor Central Boiler
* Lead Sensor warm water
* Lead Sensor Storage boiler

Outdoor temperature sensor

**Heating Circuit 3**

(Floor Heating)

* Pump 3
* Mixer 3
* Lead Sensor 3

**Heating Circuit 2**

(Living room/Wall Heating)

* Pump 2
* Mixer 2
* Lead Sensor 2
* Sensor room thermostat

**Heating Circuit 1**

(Radiators)

* Pump 1
* Mixer 1
* Lead Sensor 1

# Heating Circuit 1 - Flow chart

Difference1 > 2°C

Disable Mixer 1

Difference < -2°C

**No**

**Yes**

**No**

Enable Mixer 1

**Yes**

**No**

**Yes**

Turn-off condition met?

Enable Pump 1

Wait ca. 30 sec.

Lead sensor 1 – Determine temperatur

Determine temperatur from heating curve

1Temperature from heating curve – Lead sensor 1 temperature

**Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Turn-off conditions:**

1. Outdoor temperature > 18° OR
2. Boiler Pump is enabled OR
3. Flow temperature > 75°

**Heating curve:**

Input parameters:

1. Target indoor temperature at daytime
2. Target indoor temperature at nighttime

**Mixer duration:**

* Minimal mixer duration: 5 seconds
* Maximal mixer duration: 100 seconds
* Mixer duration is determined by temperature difference. (incremental)

**Notes:**  
In future it should be possible to adjust heating curve values at different times of the day.

# Heating Circuit 2 - Flow chart

**No**

**No**

**Yes**

Enable Mixer 2

**No**

**Yes**

**Yes**

**No**

Difference < -2°C

Disable Mixer 2

Difference1 > 2°C

Turn-off condition met?

Enable Pump 2

Wait ca. 30 sec.

Lead sensor 2 – Determine temperatur

**Yes**

3rd cycle AND lead temperature > 40°C?

**\***

Determine temperatur from heating curve

1Temperature from heating curve – Lead sensor 2 temperature

**Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Turn-off conditions:**

1. Outdoor temperature > 18° OR
2. Boiler Pump is enabled

**\*Heating Curve:**

Before determining temperature from heating curve for Heating Circuit 2 it is required to normalize the target temperature by:

Target temp. = (Target temp. - Sensor room thermostat) + Target temp.

# Heating Circuit 3 - Flow chart

**No**

**No**

**Yes**

Enable Mixer 3

**No**

**Yes**

**Yes**

**No**

Difference < -2°C

Disable Mixer 3

Difference1 > 2°C

Turn-off condition met?

Enable Pump 3

Wait ca. 30 sec.

Lead sensor 3 – Determine temperatur

**Yes**

3rd cycle AND lead temperature > 40°C?

Determine temperatur from heating curve

1Temperature from heating curve – Lead sensor 3 temperature

**Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Turn-off conditions:**

1. Outdoor temperature > 18° OR
2. Boiler Pump is enabled

# Heating Circuit 4 - Flow chart

**No**

**Yes**

**Yes**

Pump 4 enabled?

**No**

Determine temperature from central boiler

Determine temperature from central boiler

< 42°C

**No**

**Yes**

Determine temperature from storage boiler

> (42°C + hysteresis value)

**No**

**Yes**

Enable Pump 4

Difference1 > 5°C

Enable varve for district heating

1 Storage boiler temperature

**Specifications\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Notes:**

If pump 4 is enabled then the turn-off condition is met for all other heating circuits.

# Programming components

Programming language:

Ruby on Rails using RVM (<http://rvm.io>)

This command will install the latest stable versions of RVM and Rails:

$ curl -sSL https://get.rvm.io | bash -s stable --rails

I²C driver for Ruby:

<https://github.com/andec/i2c>

Database: