Wireless Programmable Thermostat Using Raspberry Pi

Soft Computing Applications pp 949-957 | Cite as

- Razvan Nicolae Vasiliu (1)
- Mircea Popa (1) Email author (mircea.popa@cs.upt.ro)
- Marius Marcu (1)

1. Computer and Software Engineering Department, Politehnica University of Timisoara, Timisoara, Romania

Conference paper
First Online: 03 November 2015

Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 357)

Abstract

This paper analyzes the reliability of a custom-made programmable thermostat using the Raspberry Pi miniature computer. The assembly uses RF sensors connected with thermistors in order to detect and send temperature information. The information is received by a base receiver connected to the GPIO port of the Raspberry board. Using a web framework and a simple algorithm, this allows elements such as a relay connected to the GPIO to be controlled via a web browser application. The proposed solution will be compared with one of the state-of-the-art devices in this field to expose the advantages of the approach.

1 Introduction

In present times, thermostats have seen an increased usage around the globe. As the technology advances, more and more households take advantage of such devices which help optimizing the energy usage along the year. The construction of these devices ranges from simple mechanical parts to most advanced electronic components which offer multiple configuration presets as well as artificial intelligence. The topic of this paper is to realize a next generation thermostat using the Raspberry Pi [1] minicomputer coupled with wireless sensors and to analyze the benefits in comparison to one of the most advanced thermostats that currently exist.

2 The Proposed Solution

2.1 Components

The central element of the proposed system (Fig. 1) is represented by a standard Raspberry Pi board which works as an interface between the sensors and the controlled relay. Temperature sensing is done by the thermistors attached to the XRF transmitters (Fig. 2). Linux, the OS that is running on the board, makes it easy to develop automation applications like the ones proposed in [2, 3].

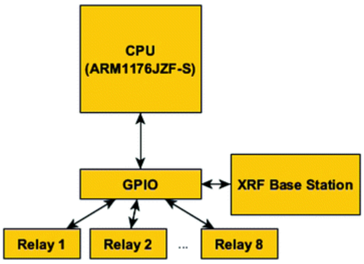

Fig. 1

The Raspberry Pi minicomputer and XRF base station

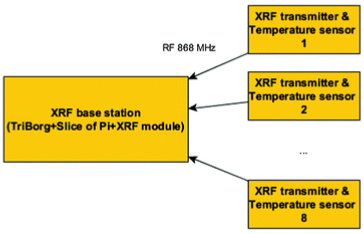

Fig. 2

The XRF base station receives data from temperature sensors through additional XRF modules

The device is based on the ARM1176JZF-S CPU running Raspbian, a Linux OS ported on Raspberry PI. Raspberry was chosen due to its small dimensions (85.60 mm × 56 mm × 21 mm) and lower energy consumption: ~2 W. Regarding connectivity, the board features 2 USB ports, 1 HDMI port, Ethernet up to 100 Mbps, an audio out and an RCS video out for analog displays. In order to operate, it requires a memory card, in our case a 16 GB SD-CARD for the OS. The 512 MB RAM onboard along with the 700 MHz ARM CPU makes it possible to run additionally relative small applications on top of the OS. A wireless nano network adapter from TP-Link was connected on the USB port to facilitate the communication with the Internet and LAN. This type of adapter was chosen also due to its small size (15 × 19 × 7 mm) and speeds up to 150 Mbps. One of the most important interfaces onboard is the GPIO port. The signals and power supplies of this port are brought out to a 26-pin header on the PCB. The ports are 3.3 V logic pins and a maximum current of 50 mA can be simultaneously drawn from all the ports. In our current setup, a Slice of Pi board is connected to the GPIO through a TriBorg port extension board. The TriBorg multiplies the GPIO three times allowing access to unused ports. The Slice of Pi uses the UART interface to program and communicate with the XRF sensor located on top of it. This uses a total of four ports including the power. Using the TriBorg extension allows to connect the relays to the remaining GPIO ports (Table 1).

Table 1

GPIO pinout for Raspberry Pi ver. B [2]