Data Collection Module for Human Activity Recognition

Dejan Gjorgjevikj
Ss. Cyril and Methodius University in Skopje Faculty of Computer Science and Engineering Skopje, Macedonia
dejan.gjorgjevikj@finki.ukim.mk

Gjorgji Madjarov
Ss. Cyril and Methodius University in Skopje Faculty of Computer Science and Engineering Skopje, Macedonia
gjorgji.madjarov@finki.ukim.mk

***Abstract*—Unobtrusive human activity monitoring using cheap and widely available sensors are the future for human activity recognition. It will support the extensive penetration of new applications in Ambient Assisted Living (AAL), Smart Homes (SH), Smart Cities (SC) and Health Monitoring (HM). The biggest challenges in these applications are the automatic processing and analyzing the large amounts of sensory data as well as building machine learning models for monitoring, detection, recognition and prediction of an activity, movement, state or an event. The aim of this research is to develop a data collection system that will enable detection and monitoring of human activity using very low-cost, unobtrusive passive infrared and microwave radar sensors. Our data collection module is composed of Arduino microcontroller, SD card module and real time clock module and enables connecting several sensors which measurements are to be logged. In our experiments we used a modified microwave radar sensor RCWL-0516 and a modified passive infrared sensor HC-SR501. Both are extremely low cost, easily accessible sensors usually used for general purpose applications like motion detection for light switching. Both sensors were modified in a way to make the raw analog output of the sensor available for logging by the microcontroller. The data collection module enables collecting measurements of up to 4 analog (10-bit precision), and up to 8 digital sensor inputs with sampling rates of up to 200 samples per second. The measurements are logged on a SD card including a precise timestamp that will enable the logs of several modules to be joined together keeping the time alignment of the readings. A separate setup for synchronized initialization of the RTC modules of the separate sensor modules is also presented. A series of experiments in a control environment with volunteers were conducted and the collected data from the sensors are pre-processed and labelled for further analysis and application of machine learning based approaches for automatic recognition and monitoring of human activity.**

***Keywords*—human activity recognition; passive infrared sensor; microwave radar sensor; Arduino; real time clock initialization**

I. Introduction

In most developed countries around the world the number of elderly people is rapidly growing as a proportion of the total population [1]. Aging typically involves decrease in mobility, strength and stamina, as well as reduced sensory acuity. Many older adults also have to deal with chronic diseases often linked to a higher rate of injuries, functional limitations, and cognitive impairments. Frailty developed as a consequence of age-related decline in many physiological parameters also leads to increased risk of falls, progressive disability, with possible need for long-term care. Also, most elderly people live at home. Frailty diagnosis is mainly based on grip strength, walking speed (gait speed) and can be assessed using self-reported and test-based health measures [2]. These parameters can be easily measured by existing devices, most of them even in an unobtrusive way monitoring the usual daily activities and may be used to monitor health status and indicators of frailty [3]. Monitoring of these parameters can be used to get preventive as well as emergency information.

Declining costs and advancements in sensor technology resulted in sensors vast proliferation that opened up unprecedented opportunities for a wide variety of industrial, scientific, commercial, agricultural and military applications, such as home automation, health care, emergency response, smart transportation, infrastructure protection, and others [4]. Utilizing new technologies and advanced signal processing methods even with the well-known measurement principles can lead to considerably improved sensor features. Also, combining the measurements from multiple sensors can significantly contribute to the enhancement of the quality and availability of information.

Human activity recognition has attracted a lot of research activity in several fields including the use of wireless sensors, positioning technologies and techniques, embedded computing, remote sensing and energy management among others. Most of the research was directed towards human activity recognition and tracking using warble sensors [5][6], as well as unobtrusive ambient sensors [7]. As wearable sensors smartphones, smart bands, and dedicated sensor nodes have been used. Among the unobtrusive sensors use of sensor nodes equipped with acoustic sensors [8], ultrasound sensors [9], pyroelectric motion detection sensors [10] and recently microwave radar sensors [11] has been reported for human activity recognition. They are considered as essential for number of enabling technologies for independent living by the elderly such as the ambient assisted living systems (AALS).

Application of such sensors in living environments, particularly when the availability of the data for monitoring, alarming and decision making purposes in an autonomous system is time critical, some kind of connectivity among the sensors and usually a centralized collection of all the measured data is needed calling for full featured data acquisition systems. However, for some applications when the primary purpose is exploring the possibilities of using a different setups of sensors, sampling and data collection strategies in controlled or semi-controlled environment with primary purpose of further off-line data analysis or collecting and labeling data for machine learning approaches, autonomous independent sensor nodes capable of recording internally the measurements may be a valuable option. Such devices are usually known as data loggers and can be preferred option especially in experimentation and research phase. However typical data loggers usually target monitoring of slow changing physical phenomena on the range of sampling rates of 1 Hz or lower.

The aim here was to develop high capacity data collection module that will be able to record readings from several sensors at moderate sampling rates in the range of at least 100 samples per second for a prolonged time. Also considerations should be taken that recordings of several such independent modules should be easily combined in a time aligned fashion, also aligned to video footage for further off-line analysis and labeling of the events. In our approach we have used such data collection modules for collecting measurements from several sensors for human activity recognition. The proposed data collection platform itself is more general and can be easily configured and used for other purposes.

The remainder of the paper is organized as follows: in Section 2 we describe the principles of the proposed system for data collection. In Section 3 the hardware and the software of data collection system we have developed is presented and discussed. Finally, we conclude in Section 4 with remarks on our future work.

II. System Overview

We have developed a very low cost data collection module around the Arduino platform that can collect the sensor readings and store them on a SD card for further processing. The module integrates three primary components: a microcontroller board, a real-time clock (RTC) module and a microSD card adapter for storing the data on a secure digital memory card. A number of different types of sensors can also be attached to the microcontroller for measuring different parameters. In particular, in our experiments we use motion detection sensors, namely passive infrared (PIR) sensor and microwave radar sensor. The whole setup can be powered by a battery or an appropriate DC power adapter. Two light emitting diodes (LEDs) were used to indicate the status of the module.

Appropriate software for the microcontroller was also developed in the Arduino environment that enables measuring several (up to 4 analog values with 10-bit precision, and up to 8 digital sensor inputs) supporting rates of up to 200 samples per second. The sensor measurements and also the current temperature (obtained from the RTC at a lower rate) are stored on a microSD card together with a precise timestamp of the measurements so that the measurement taken by several independent nodes can be easily combined keeping them accurately time aligned.

Taking into account that we have used sampling frequencies in the range of tenths to hundredths of hertz, the precision and the synchronization of the real time clocks in the separate modules turned to be very important for accurate time alignment of the collected measurements. Therefore a separate setup around the NodeMCU board (utilizing the ESP8266 microcontroller) that enables connecting of up to 4 RTC modules and their synchronization to millisecond precision was developed. Appropriate software that enables setting the time and some other internal parameters of the real time clock modules was also developed.

III. Hardware Design and Interfacing

The new very low-cost portable data collection platform is presented in this section, including both hardware and software. The microcontroller board manages the other modules and sensors by utilizing several data protocols.

A. Data Collection Platform and Principles of Operation

Having the application area in mind, the desirable characteristics for our data collection platform would be:

- Built from a small number of low-cost, readily available components;
- Designed and constructed using open source hardware and software;
- Support for wide variety of analog and digital sensors
- Removable microSD storage media
- Being able to store as much data as possible on the memory card meaning that it can record at higher sampling rates for long time before the memory is exhausted
- Easily adjustable operating parameters, such as sampling frequency, number and type of sensors
- Precise timekeeping and timestamping, with low jitter

Arduino is an open-source electronics prototyping platform based on easy-to-use hardware and software. It is very cheap and easily accessible and well supported by an open community of developers. One of the greatest strengths of the Arduino is the cross-platform Integrated Development Environment (IDE) which presents a simplified C++ programming interface that leverages extensive code libraries supporting most mainstream sensors.

We have used Arduino Uno and Nano boards for setting our data collection modules. Their hardware is based around the ATmega328 microcontroller that has 14 digital input/output pins, 8 analogue inputs (only 6 of them are accessible on the Uno board), and include a USB connection used for programming the microcontroller that can also power the board. The boards can operate using an external power supply (USB cable, AC/DC adapter or a battery) of 7V to 12V.

The connection diagram of the whole setup is shown on Fig. 1. The Arduino Uno microcontroller board is interfaced to the real time clock module through an I^2C bus and to the microSD card module through a SPI bus. The real time clock is responsible for time keeping that is used for logging but also for precisely timed interrupt generation according to which the sensors are sampled.

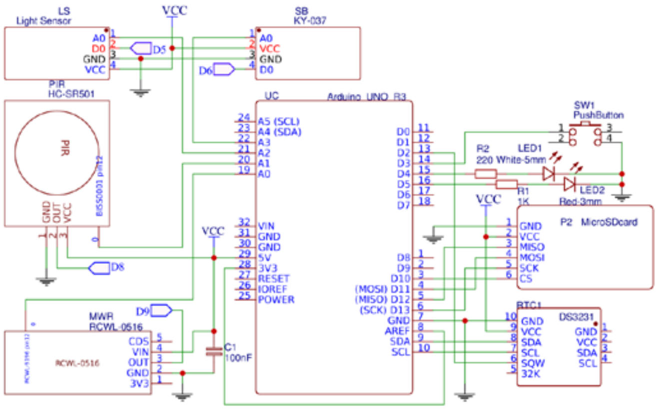

Fig. 1. Data collection module schematics

The Arduino's 10-bit single-ended analog-to-digital converter (ADC) provides 1024 sampling levels between a common ground and the reference voltage that can be configured to be the microcontroller supply voltage, internal 1.1V reference or externally supplied reference voltage. Since two of the analog pins (A4 and A5) on the Arduino are reserved for inter-integrated circuit (I^2C) bus communications with the RTC and possibly other sensor modules, this leaves 6 pins available for use with analog sensors (A0–A3, A6 and A7). On the Arduino Uno boards the pins A6 and A7 are not broken out and are not directly available.

Most 328P-based Arduinos have 13 digital I/O pins, 9 of which are used in this data logger, leaving 4 available for digital inputs that can be logged. Pins D0 and D1 are typically reserved for serial outputs through a Universal Asynchronous Receiver-Transmitter (UART). Hardware interrupt pin D2 is connected to the RTC's programmable alarm and governs the accurate timing for sampling the sensors. A pushbutton that connects D3 to ground is used for simple configuration during boot up. Two indicator LEDs (red and white) are connected to D4 and D5 through current limiting resistors. Four digital pins are used for SPI communications with the microSD card: D10 as a Cable Select (CS) line, D11 for Master-Out-Slave-In (MOSI), D12 for Master-In-Slave-Out (MISO) and D13 for the bus clock line (SCL). This configuration leaves 4 digital pins available for sensor input (D6–D9).

The oscillator circuits on most Arduino boards are not very accurate and have significant variation due to thermal effects. Therefore, it is not very reliable to use the microcontroller clock for timing, especially in the circumstances where one would like to keep the sampling on separate boards reasonably synchronized for some time. Because of this, in our approach, we are using a real time clock module for accurate time keeping. The module integrates DS3231 chip from Maxim, a lithium coin cell backup battery for continuous timekeeping and a 4 kB AT24C32 EEPROM.

The DS3231 is very low-cost and extremely accurate real-time clock with an integrated temperature-compensated crystal oscillator and crystal. It provides ±2ppm accuracy that is around ±1 min drift per year. It also integrates a digital temperature sensor with 0.25ºC resolution that is used internally for the oscillator correction but can also be read from the RTC's data registers [3]. The RTC module keeps track of the current time and date and can be programmed to produce alarms (raise a signal on one of its pins) at predefined time or frequency. The RTC can be configured to interrupt the microcontroller every second, 1024, 4096 or 8192 times a second.

The sensor measurements are stored on the SD card on regular intervals. The microSD card module is used as interface to a SD card used for permanent data storage. It integrates a microSD card slot, a power regulator and level shifting circuit. The SD card can then be simply removed so that the data can be easily transferred to a computer for further analysis.

B. Sensors

In our experiments we have used motion detection sensors, namely the passive infrared (PIR) sensor module HC-SR501 and the microwave radar sensor module RCWL-0516. These are very chip sensors for general purpose applications and are mainly used for motion detection light switching. They both have only single digital output that is active when motion is detected (and is being kept active for certain time afterwards) that is usually used to switch other devices. We have modified both sensors in a way that the analog pre-amplified raw sensor signal was taken from the dedicated motion detection IC already present on the modules and fed to the analog input of the microcontroller.

1) Pyroelectric infrared (PIR) sensor

PIR sensors are passive sensors that detect temperature differences. They are suited to detecting the motion of people by their body temperature and are widely used as a presence trigger. However, the analog output of PIR element depends on several other aspects, including the distance of the body from the PIR sensor, the direction and speed of movement, the body shape and gait [2]. They use a special Fresnel lens to generate multiple thermal images of a warm object, such as a person. The sensor compares the intensity of the infrared rays emitted by bodies at body temperature from time to time. If there is an object movement, then the intensity changes which results in detecting the object.

HC-SR501 PIR sensor is low cost device that is very easily accessible (Fig. 2). This motion sensor module uses the LHI778 Passive Infrared Sensor and the BISS0001 IC to control how motion is detected. The module features adjustable sensitivity and can detect movement in a 110º cone shape space at distances between 3 and 7 meters. The module also includes time delay adjustments and trigger selection that allow for fine tuning according to its application.

BISS0001 is a PIR controller, hybrid analog/digital IC manufactured in low power CMOS technology and it requires very few external components for its application circuit [4]. It incorporates several high input impedance operational amplifiers, bi-directional level detector, built-in power up disable and output pulse control logic and supports both retriggerable and non-retriggerable mode of operation. The inputs and the outputs of two of the integrated operational amplifiers are broken out on the pins of the IC so that they can be configured for amplification and filtering of the PIR element signal by external passive components. The pin 12 denoted as 2OUT is particularly interesting as it is the output of the second stage amplifier that is further compared to reference voltages and drives the control logic and the delay timer that produces the digital output used for turning the external devices on or off. Instead of designing own circuitry around a PIR element with operation amplifiers to produce analog signal powerful enough to be read by a microcontroller, we have taken the analog signal out of the pin 12 of BISS0001 (Fig. 4) and have led it to an analog pin of the microcontroller in order to obtain the raw analog signal of the detected motion that will be sampled and logged by the microcontroller for further processing and analysis.

Fig. 2. Passive infrared sensor module HC-SR501

2) Microwave radar sensor

Microwave radar sensors emit microwave signal and compare the frequency of the echo with the emission frequency listening for a Doppler shift from a moving object. The antennas of such devices are usually low cost, die cast metal horns. RCWL-0516 (Fig. 3) is a radar microwave motion sensor module which can act as an alternative to a PIR motion sensor. The module is extremely low cost module with minimalistic design that incorporates a single high frequency NPN transistor MMBR941M that acts as microwave frequency transmitter, receiver and mixer, some passive components and distinctive PCB design which traces act as antenna, inductivities and capacitances that brings the transistor to self-oscillation of about 3.1 GHz [5]. The PCB also incorporates RCWL-9196 IC that is almost identical in function and pinout to BISS0001.

Fig. 3. Microwave radar sensor module RCWL-0516

The low frequency difference is extracted by a low pass RC filter and amplified by the RCWL-9196 IC and treated exactly the same as a signal from a PIR sensor. There are very similar low cost radar motion sensor modules on the market that are actually using the BISS0001 rather than the RCWL-9196 IC. We have also taken the analog signal out of the pin 12 of RCWL-9196 (Fig. 4) and led it to an analog pin of the microcontroller for logging.

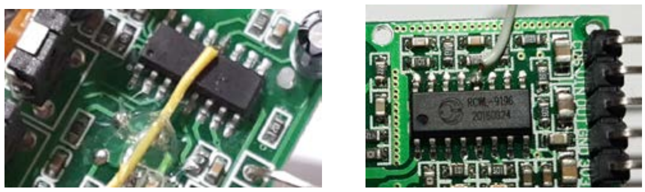

Fig. 4. A wire soldered directly to pin 12 of BISS0001 (left) and RCWL-9196 (right) to take the analog sinal to the microcontroller

Unlike the PIR sensor that is directional and reacts on moving "warm" objects that are emitting infrared of certain wavelets, the radar microwave sensor is omnidirectional and reacts on moving metal objects and objects that contain high percentage of water (humans and animals) on a distance of about 7m. It is important to mention that the radar sensor can also detect movement even when the moving object is behind an obstacle, even through thin walls, but can suffers from interference by other moving objects like fans. The digital outputs of both sensors (especially the microwave one) are very sensitive to variations in the supply voltage, so when the digital outputs are used, installing a filter capacitor near the module is recommended [6].

As for the BISS0001 and RCWL-9196, they both internally operate on 3.3V (RCWL-9196 IC has integrated 3.3V regulator, while HC-SR501 module incorporates a HT7133 voltage regulator) and are configured in a way that the output of the second amplifier is at about half of this voltage when idle (no movement detected) which makes it suitable for direct sampling by the microcontroller. As we were using Arduino microcontrollers that were operating on 5V, in order to increase the resolution of the built in AD converter in the microcontroller, we have configured it to use 3.3V as external reference voltage for the AD conversion. This way the effective resolution of the ADC becomes 3.22 mV/bit.

3) Light and sound sensors

In our setup, we are also using two other modules that can be used for presence detection, a sound detection sensor and a light sensor. The low cost microphone sound detection sensor module KY-037 has an analog output offering the amplified signal from the microphone suitable for sampling by microcontroller and a digital output that outputs high level when the sound intensity reaches a certain threshold that can be adjusted via potentiometer on the sensor. The light sensor module uses a LDR-resistor (Light dependent resistor) and also provides analog and digital signal output. The analog signal is higher when it is darker and the digital signal becomes active when the ambience light falls to a certain level that can be adjusted via potentiometer on the sensor.

4) Sensors connectivity

In our experiments the RCWL-0516 radar microwave sensor, HC-SR501 PIR sensor, a light sensor module and microphone sound sensor module were connected to the data collection module. The analog outputs of the sensors are connected to the pins A0-A4 on the Arduino and the digital outputs of the sensors to the pins D9-D6.

C. *Data collection module software overview*

The software in the data collection modules was developed in Microsoft Visual Studio 2015 using the Visual Micro plugin that supports developing Arduino compatible cross-platform programs for number of different Arduino compatible micro-controllers. Its basic responsibility is to take the sensor readings (sample the signals on the microcontroller inputs) on regular intervals and write those on a permanent media device (the SD card) together with a timestamp of the current time as obtained from the RTC module. The software can be configured for different number of sensors and different sampling rates and logging strategies. It supports logging in compact binary format or standard coma separated values (CSV) textual format.

The RTCtime Arduino library by Sergio Manzi [7] and William Greiman's SdFat Arduino library [8] were used for accessing the functionality of the DS3231 RTC and storing data on the SD card. The program that is executed by the microcontroller in each data collection module basically performs the following tasks:

Initialization:

- Configure the microcontroller ports and set the analog reference
- Check if the pushbutton is pressed and change the logging mode from binary to textual if pressed
- Initialize the RTC and configure it to interrupt the microcontroller 1024 times a second. Also initialize the SD card and if either initialization has failed, go to an endless cycle while the red status led is emitting a distinctive combination of flashes according to the cause of the failure.
- Write a record in the SENSBOOT.LOG file on the SD card with the timestamp of the current time when the node has been turned on and its configuration parameters (sampling rate and logging format).
- Open log file with name SENSmmdd.BIN or SENSmmdd.CSV according to the chosen format where mm and dd denotes the current month and day. If the file already exists sensor readings will be appended on the end of the file.
- After waiting a predetermined time (flashing both LEDs) for the sensors to stabilize, the interrupt handler is initialized and the code enters the main loop that repeats until it is turned off or an error is detected.

Loop:

- Every 41st interrupt (about every 1/25 seconds) read the sensor values and store them on the microSD card. If logging in compact binary format, before each predetermined number of samples (338 in our setup, but can be configured) write a header with a timestamp and the current temperature measurement. The header is followed by a 338 6-byte entries holding all sensor measurements without timestamp for each of them as they are sampled on interrupt driven regular intervals.
- After writing predetermined number of samples flush all previous writes on the SD card. This way no more than the last several seconds of measurements can be lost if the node is turned off. Every flush on the SD card is accompanied by a short flash of the red LED. Flushing the writes on the SD card is slow operation that is not recommended after writing every sample when logging on higher rates.
- Every 10 seconds flash the white led. These flashes are used for monitoring synchronization of the nodes and are meant to be easily noticeable on a camera footage that could be used for precise labeling of the events to the recorded data. The frequency and the duration of the flash can be configured.
- Continuously: The red LED acts as a status indicator emitting breathing like modulated signal during logging. If an error is encountered during logging (communicating the RTC or the SD card module or the microSD card memory is exhausted) the program goes into endless cycle emitting distinctive combination of flashes of the red LED according to the cause of the failure. Pressing the pushbutton toggles echo to serial. If echo to serial is enabled all sensor reading values in CSV format are sent to the serial monitor.

Writing the data in binary format can save considerable space on the SD card. For example, storing 4 analog (10 bit precision) and up to 8 digital values takes only 6 bytes per sample plus a 20 byte header every few seconds. Compared to over 50 bytes per sample if CSV format is used, this is considerable saving in storage space on SD card. Storing the data in binary format is also faster, so higher sampling rates can be achieved. The use of the compact binary format enables continuous logging with 100 samples per second for over 2.5 months before a 4GB SD card runs out of space.

D. *Synchronizing the clocks of the RTC modules*

Because the data collection modules were operating independently and were not connected to each other, logging on their own on a rather high frequency, the accuracy and the synchronization of the time presented by the RTC was important. Relaying solely on the microcontroller cycles for determining the time when to take a sample proved to be very inaccurate since the crustal oscillators used in cheap microcontrollers like Arduino are not precisely calibrated and are not temperature compensated. While this difference in oscillator frequencies at 0.1% does not look significant for general purpose applications, it means that in 10 minute logging session at 25 samples per second, one module can produce over 20 samples more than another, which complicates the alignment of the measures.

However, although the RTC is more accurate, it provides time to a second precision only. Fortunately, it can be configured to produce alarms on its SQW pin that can be used as an interrupt to initialize pooling the sensors on regular intervals. In our setup we have configured the RTC clock to interrupt the microcontroller 1024 times a second, and the microcontroller to pool the sensors on every 41st interrupt (roughly 25 times a second).

In order to initially synchronize the clocks of the RTC modules a separate setup around the ESP8266 microcontroller on a NodeMCU board was developed. One of the challenges was to connect several RTC modules on a same microcontroller. Since the DS3231 RTC is device that communicates using an I^2C bus with the microcontroller and has a preconfigured address that cannot be changed, only one such device can be connected on I^2C bus. In order to connect several RTC modules to a single microcontroller for synchronization we have to configure several separate I^2C buses on a single microcontroller. As the ESP8266 does not have hardware implementation of the I^2C bus, any pair of pins can be used for software version of the I^2C protocol using bit banging. As for setting the clocks and their parameters only we do not need any interleaved or parallel access to all of them in the same time, we can even share the clock signal among all the modules. This way we have saved some pins for connecting an 8 digit 7-segment LED display that can show the current time including hundredths of a second.

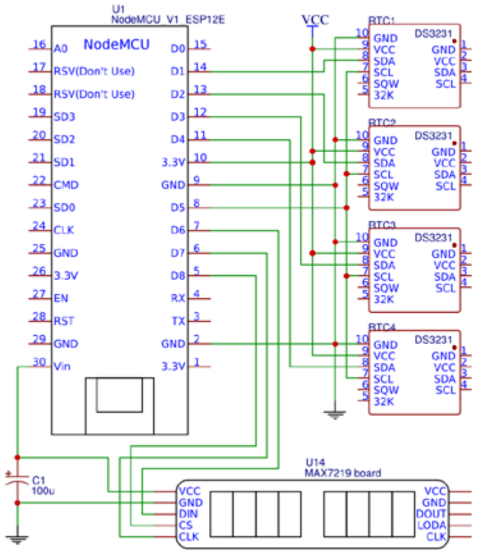

Fig. 5. RTC modules clocks synchronization schematics

In order to enable accessing several RTC modules on the same microcontroller we have developed a RTC library (based on the RTCtime Arduino library) that allows simple addressing of multiple RTC clocks, each configured with object representing the software implementation of the I^2C protocol on particular pins.

Setting the parameters is performed using the serial UART interface of the microcontroller that takes the commands typed in the serial terminal, interprets them and executes them. Since the ESP8266 has Wi-Fi capabilities and incorporates a TCP/IP stack, we have implemented the possibility for the microcontroller itself to connect to a NTP server and request the current time. For connecting to a Wi-Fi access point the SSID name, the password, and also the NTP server address and the local time zone has to be specified (also using simple commands from the serial interface). These settings are stored in the flash memory of the microcontroller in order to be persistent along switching off the setup. Also, the program allows setting the current date and time (entered on the serial terminal as part of the command) to a specified RTC or all the connected RTC modules at once. Possibility to set the seconds only, synchronize the time of all the RTC modules to a specified one, display the temperature measured by each of the RTC modules, reading and setting the value of the aging offset register for each RTC module, enabling/disabling the clock on the 32K pin (used for calibration) and setting the brightness of the LED display is also provided.

This setup (Fig. 6) was used to plug the RTC modules and synchronize all their clocks to a common time. After that each of the RTC modules is plugged in one of the separate data collection modules for recording human activity. The first RTC module remains connected to the board for showing the current time on the display that is recorded by a camera. The time shown on the display of the synchronization module (also captured by the camera) is used to determine the offset of the time embedded on the camera recording, because the camera clock can be set only manually.

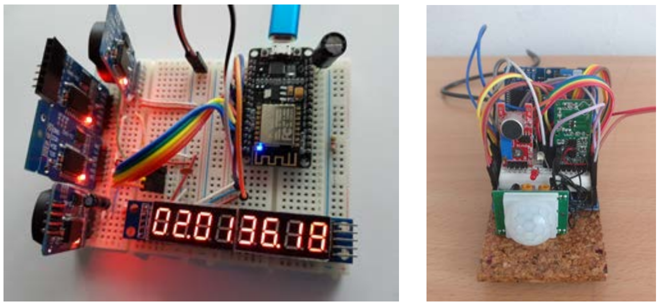

Fig. 6. RTC modules clocks synchronization setup (left) and human activity data collection module (right)

E. *Recording Human Activities*

Three data collection modules (two around Arduino Nano and one around Arduino Uno) were prepared connecting all the components using a breadboard and jumper wires (Fig. 6). Two of the modules were placed near the side walls and one was placed on the ceiling near the center of the room (Fig. 7).

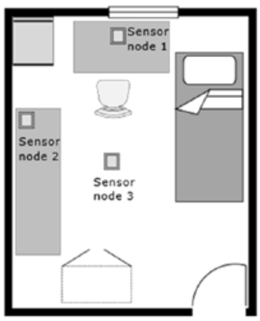

Fig. 7. Floorplan of the room wher the sensors were placed

A group of volunteers were instructed to perform different activities in random order while the signals received from the sensors were recorded. The activities include: sitting down on a chair, standing up from the sitting position, walking around the room, laying on the bad, standing up from the bad, walking out of the room and walking in the room. The activities were also recorded by a dashboard camera in order for the footage to be used for offline labeling the start and the end in time of each different activity for machine learning purposes.

Recorded activity of 20 seconds sampled at 25 samples per second by the 3 independent data collection modules each logging 4 analog sensor values is shown on Fig. 8. The activity corresponds to entering the room and sitting down on the bad.

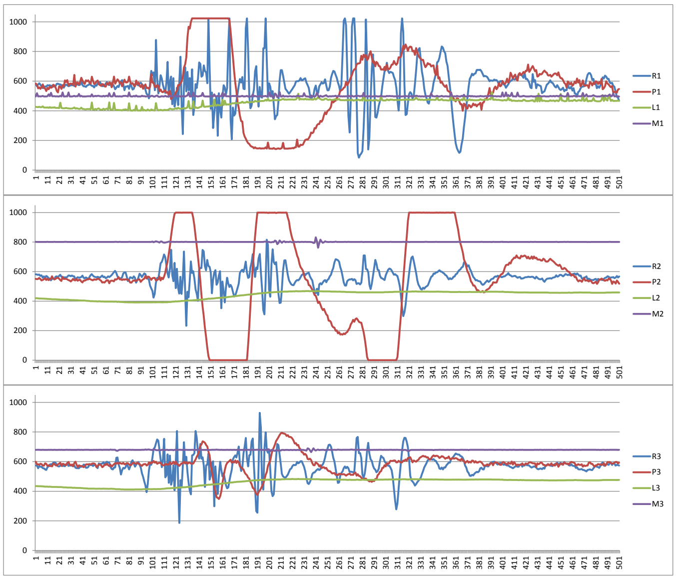

Fig. 8. Sensor mesurement of human activity by the 3 data collection modules. The letter reperesents the sensor (R-radar, P-PIR, L-light, M-mcrophone) and the digit corepondes to the sensor placement

IV. Conclusion

A very low cost data collection system around the Arduino platform that enables detection and monitoring of human activity using unobtrusive sensors was developed. The module enables recording of the sensor readings using sampling frequencies of up to 200Hz and storing them on a micro SD card. We have modified two very cheap easily available motion detection sensors in order to obtain analog signal from them. For precise time keeping and keeping the measurement of the independent modules synchronized a precise real time clock was used. A separate setup around the ESP6266 microcontroller was developed for initialization and synchronization of the RTC modules of the separate data collection modules. A limited number of experiments were performed recording activities of group of volunteers in a controlled environment. The obtained sensor data will be annotated and used for developing machine learning approaches for human activity recognition using unobtrusive sensors.

Acknowledgment

This research was partially funded by the Faculty of computer science and engineering, Ss. Cyril and Methodius University in Skopje.

References

[1] "World Population Ageing 2015", (ST/ESA/SER.A/390), United Nations, Department of Economic and Social Affairs, Population Division (2015).

[2] O. Theou, M. O'Connell, B. King-Kallimanis, A. O'Halloran, K. Rockwood and R. Kenny, "Measuring frailty using self-report and test-based health measures", Age and Ageing, vol. 44, no. 3, pp. 471-477, 2015.

[3] H. Saner, "Wearable Sensors for Assisted Living in Elderly People", Frontiers in ICT, vol. 5, 2018.

[4] O. Kanoun and H. Trankler, "Sensor Technology Advances and Future Trends", IEEE Transactions on Instrumentation and Measurement, vol. 53, no. 6, pp. 1497-1501, 2004.

[5] O. Lara and M. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors", IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1192-1209, 2013.

[6] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou and Y. Amirat, "Physical Human Activity Recognition Using Wearable Sensors", Sensors, vol. 15, no. 12, pp. 31314-31338, 2015.

[7] E. Salomons, P. Havinga and H. van Leeuwen, "Inferring Human Activity Recognition with Ambient Sound on Wireless Sensor Nodes", Sensors, vol. 16, no. 10, p. 1586, 2016.

[8] J. Sim, Y. Lee and O. Kwon, "Acoustic Sensor Based Recognition of Human Activity in Everyday Life for Smart Home Services", International Journal of Distributed Sensor Networks, vol. 11, no. 9, p. 679123, 2015.

[9] J. Sim, Y. Lee and O. Kwon, "Acoustic Sensor Based Recognition of Human Activity in Everyday Life for Smart Home Services", International Journal of Distributed Sensor Networks, vol. 11, no. 9, p. 679123, 2015.

[10] X. Luo, Q. Guan, H. Tan, L. Gao, Z. Wang and X. Luo, "Simultaneous Indoor Tracking and Activity Recognition Using Pyroelectric Infrared Sensors", Sensors, vol. 17, no. 8, p. 1738, 2017.

[11] G. Diraco, A. Leone and P. Siciliano, "A Radar-Based Smart Sensor for Unobtrusive Elderly Monitoring in Ambient Assisted Living Applications", Biosensors, vol. 7, no. 4, p. 55, 2017.

[12] J. Yun and S.-S. Lee, "Human movement detection and identification using pyroelectric infrared sensors", Sensors, vol. 14, no. 5, pp. 8057–8081, May 2014.

[13] Maxim Integrated, "Extremely Accurate I²C-Integrated RTC/TCXO/Crystal", DS3231 datasheet, Rev. 10; 3/15, 2015. https://datasheets.maximintegrated.com/en/ds/DS3231.pdf. [Accessed: 17- May- 2018]

[14] Silvan Chip Electronics Tech.Co.,Ltd., "Heat-Induction Sensor Circuit", BISS0001 datasheet, http://www.sc-tech.cn/en/BISS0001.pdf. [Accessed: 17- May- 2018]

[15] J. Desbonnet, "Information about RCWL-0516 microwave proximity switch module (ICStation.com SKU 10630)", GitHub repository, 2017, https://github.com/jdesbonnet/RCWL-0516. [Accessed: 17- May- 2018]

[16] R. Clark, "Investigating a RCWL 9196 / RCWL-0516 "Radar" motion detector module", Roger Clark, 2017. [Online]. Available: http://www.rogerclark.net/investigating-a-rcwl-9196-rcwl-0516-radar-motion-detector-modules/. [Accessed: 17- May- 2018]

[17] S. Manzi, "A time.h compliant library that makes using the DS1307 and DS3231 Real Time Clock modules really simple", GitHub, 2017. [Online]. Available: https://github.com/smz/Arduino-RTCtime. [Accessed: 17- May- 2018]

[18] B. Greiman, "Arduino FAT16/FAT32 Library", GitHub, 2013. [Online]. Available: https://github.com/greiman/SdFat. [Accessed: 17- May- 2018].