SENSOR ENABLED SMART SUIT ELECTRONIC IOT DESIGN PLATFORM

Migdat Hodzic[1,2], James M. Brennan[2], Enis Dzanic[3]

[1] American University in Bosnia and Herzegovina, Sarajevo, Bosnia and Herzegovina
[2] BH Analytics, USA and Sarajevo, Bosnia and Herzegovina
[3]University of Bihac, Bihac, Bosnia and Herzegovina

KEYWORDS: Sensors, IOT, Smart Suit, WiFi, Firmware

ABSTRACT :

An integrated smart suit sensor and positioning system electronic IoT prototype has been developed to address the growing need for personal welfare monitoring of first line responders, defenders, and workers exposed to industrial or other hazards, as well as othere commercial and defense applications.. The system provides a GPS position map with coordinate data, current GMT time readout, subject's heart rate, body temperature, and a long-wave thermal video camera that provides a forwardlooking thermal image. Physiologic data and thermal imaging of the subject may be viewed by monitoring personnel using Internet browser connected to the system's static IP address. The system is Wi-Fi connected to a local network, which can be extended to enable secure connection to the Internet with incorporation of additional firmware. Details regarding hardware and software configuration are presented along with an appendix containing additionbal data. Ssource code for the software modules currently running on the prototype system are also available for interested parties or potential users and customers.

1. INTRODUCTİON

Recent years witnessed very considerable development in the areas of various sensors for many related applications. In this context new IoT (Internet of Things) technologies emerged as a response to a growing need to conect variety of devices in our homes, in the streets, cities, or sensor enablewd devices which attach to our body or uniforms. New low power wireless and wired sensor technologies have been developed and are used more and more in many old and lots of new applications [12]. Sensors range from environmental, physiologic, range measurements, proximity sensors and all the way to very sophisticated special purpose sensors for industrial and defense use [6, 7, 8]. Varity of embedded and inexpensive platforms exist now for fast design and prototyping such as Raspberry Pi [2]. Besides new sensor and IoT technolgies, new „smart“ materials are also becoming available more and more, with a variety of functionality built in, even some basic electronics built into the material fabric [9]. One specific and important area of sensor development are IR and thermal sensors and cameras based on them [1]. These sensors now allow for very detailed IR or thermal image sensing and digital framing with usable number of video frames [1]. In any useful IoT sensor enabled smart suite there is typically a need for positioning data and hence GPS sensors are also required. In the smart suite case, it would be required to have physiological data of the person wearing the suit, and this means a need for temperature, heart rate, blood pressure, outisde environment pressure, humidity and temperature, sensor for some dangerous gas presence (such as in mines). It is in this spirit that we developed simple IoT sensor based smart suit design and testing electronic platform described in this paper. We opted to incorporate only basic sensor components such as GPS with the antenna, temperature and heart rate sensors, as well as thermal sensor. The platform is based on popular Rapsberry Pi HW and SW computer board, with WiFi and 4G built in for Internet communications. In order to be able to demonstrate smart suit design platform we also incorporated ability to view the suit on line using its static IP address. WiFi is used for Internet connectivity. A variety of customizations are posssible depending on the interest of final users and customers and their needs for specific sensors.

2. SYSTEM COMPONENTS

A general system block diagram of the Smart Suit System electronics is shown in Figure 1. More specific component choices for our prototype and demonstration design are indicated in. Figure 2. The thermal camera (FLIR Lepton brand) is connected to host processor via SPI0 bus. The subject’s heart beat and surface body temperature sensors are connected to host processor through the Analog to Digital converter (ADC) via SPI1 bus and processed by an algorithm to extract heart rate and average temperature. Additional inputs to the ADC are available for future expansion capabilities to provide respiration rate and activity monitoring. The Host Processor for the Smart Suit System is a Raspberry Pi 3 running a Debian Stretch Linux distribution. When the system powers up it automatically starts a C++ based thermal imaging application as well as a Python based Flask Webserver, which also inputs and formats all incoming data for presentation as a served webpage.

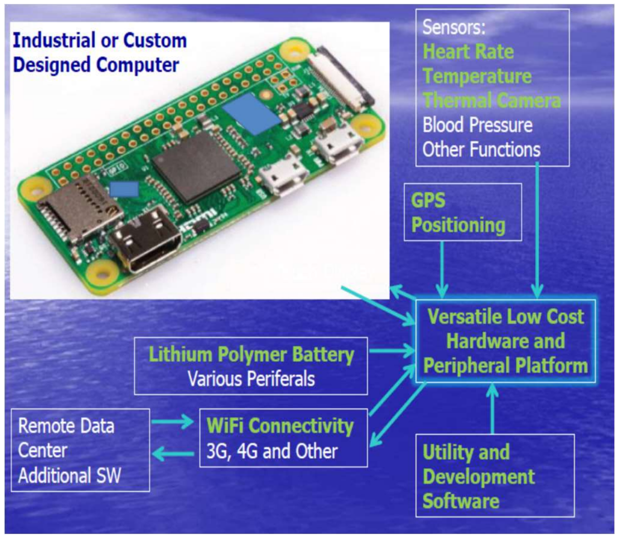


Fig. 1. Smart Suit System general electronic block diagram

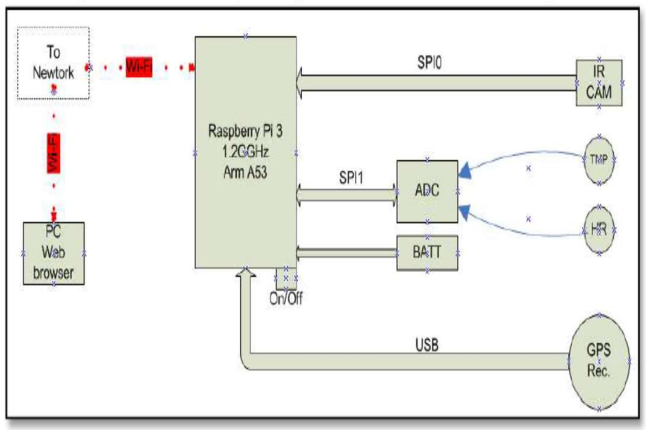


Fig. 2. Smart Suit System specific block diagram

2.1 FLIR Lepton IR camera

The FLIR Lepton is an infrared camera system that integrates a fixed-focus lens assembly, an 80x60 long-wave infrared (LWIR) microbolometer sensor array, and incorporates signal processing electronics [1]. Easy to integrate and operate, Lepton is intended for mobile devices as well as any other application requiring very small footprint, very low power, and instant-on operation. Lepton can be operated in its default mode or configured into other modes through a command and control interface (CCI). Using its default conditions, the FLIR Lepton camera outputs 60 video packets per frame, each 1312 bits long, at approximately 25.9 frames per second. The minimum output data clocking rate to the camera is on the order of 2 MHz to allow it to keep up with its real-time image generation. The camera data output provides three repeated identical frames in a row, followed by a new frame making up another series of three frames. This activity will repeat indefinitely if not interrupted or an error has happened. It should be noted that this frame series format means that real-time new frame output is approximately at a 9 Hz rate so that actual frame processing only operates using only one out of three frames. The Smart Suit System processes thermal frames at about nine per second but, at present, only grabs one still frame per web page update. Software processing of the thermal frames includes false color map encoding that normalizes the brightest pixels to the most intense coloring. This insures that the image does not saturate thus masking less bright features of the scene.

2.2 IR Camera Software

Image processing is done in the C++ language and includes algorithms to identify and synchronize processing starting with frame packet number one, formatting and processing each FLIR Lepton data line input into a 480 x 800 pixel image frame, color mapping the brightness of pixels, then converting the frame into a JPG format for storage in memory. The frame stored in memory is accessed by the

Python-based web server application approximately once every two to three seconds for output to the client display for the demonstration purposes.

2.3 Python based Flask Web Server

A Flask micro web server has been implemented to serve a web page containing human subject physiologic data, current GPS coordinate data, an up-to-date GPS location map, and the IR thermal image showing a color mapped scene ahead of the subject. This server is implemented in Python, which generates a new web page when a client application asks for an update (currently once every 2 to 3 seconds). Processing includes capturing GPS time and coordinates, acquiring subject's heart rate and external body temperature, computing and formatting all of these values into a python dictionary, and then sending the dictionary (data strings) to the client web page Client Webpage View. The client web page presently uses a timed refresh period to ask for an update from the server every two to three seconds. The data is collected in Java variables within the page and processed to form result text, a google API map of location, and to render the thermal JPG image seen in Figure 3 (Sarajevo, Bosnia an Herzegovina location). At this point we did not spend lots of time in making GUI more sophisticated. As Figure 3 indicate the GUI is just a basic one for the suit system demonstration and prototyping purposes.

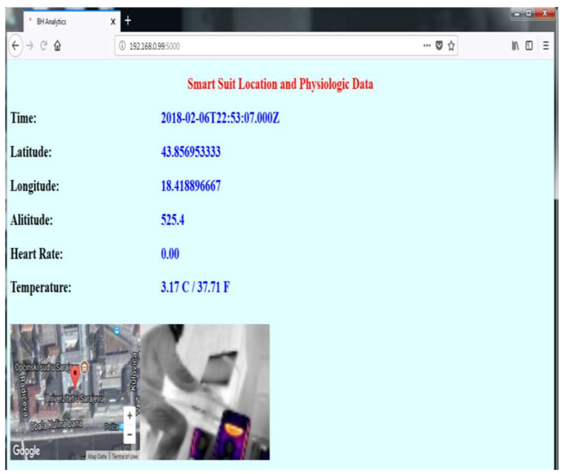


Fig. 3. Client web page displaying collected data, map, and image located in Sarajevo, B&H

2.4 Raspberry Pi 3 Debian Stretch Operating System Start

The Smart Suit System connection diagram shown in Figure 4 relies upon the underlying Linux operating system to host the thermal and Flask web server applications. At power-on the system automatically starts both the Flask server and the Lepton Thermal Imager applications. The Flask web server startup process uses the CHRON daemon to read a script at startup. This is initially setup during development using the CRONTAB command and is done from within the directory **/var/spool/cron/crontabs/pi**. See Appendix for the general procedure. During startup the operating system will look for shell scripts in the user's home directory, in this case **/home/pi/startup.sh**. The script found at this location is used to start up the Lepton thermal imaging application. Content of startup.sh can be obtained if a customer requires this. It should be noted that the Lepton thermal imaging application named "Demo", as found in directory **/home/pi/Qt/Demo/build-Demo-kit2-Debug**/, is a stand-alone executable and has been built using the Qt4 development application. Setup of the Qt4 environment is referenced in smart suit system documentation.. Another script that is automatically executed during the startup process allows system shutdown when a specific hardware pin is grounded. This script is found at **/etc/init.d/listen-for-shutdown.sh** and runs a Python script found at **/usr/local/bin/listenfor-shutdown.py**, which when invoked runs continuously in the background waiting for a hardware pin instituted shutdown interrupt signal. Once the system starts up and the required applications are each separately running, the Flask web server is ready to respond to a client application request if the Wi-Fi feature is operational.

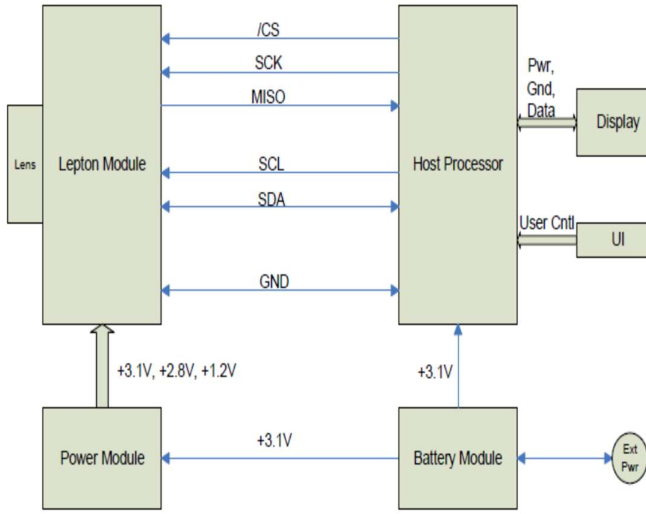


Fig. 4. Smart Suit System Connection Block Diagram.

3. SYSTEM HARDWARE

The connection diagram in Figure 6 shows the Raspberry Pi 3 I/O ports and general pins used to interface system hardware components. Numerous connectors are used to route sensor data to the required I/O pins. Depending on number of required sensors, the I/O pins functionality can be customized for a specific application. The system has enough capacity to acmmodate number of additional sensors if required.

3.1 Hardware Data Collection and Transfer

The Raspberry Pi 3 (RP3) uses 2 separate SPI busses to transfer data: 1.) SPI0 to clock in data from the MCP3008 ADC; and 2.) SPI1 to clock in data from the FLIR Lepton thermal imaging module. This is done to provide the different clock speeds required by each device. The MPC3008 ADC uses a 200KHz transfer clock to provide the lower demand heart rate and temperature measurements, which are collected at 10 samples per second. The FLIR Lepton thermal module, however, requires a minimum transfer clock of 2MHz to provide its 27 fps image rate. The thermal module operates in an autonomous manner in its present configuration after it is powered on by providing serial data in an open loop fashion when given a clock. No other programming is required to access its data via SPI serial bus. At present a momentary power interrupt switch is placed in the thermal module’s Vdd input to allow for operator reset of the device due to over temperature or other uncontrolled noisy conditions. A later version of the Smart Suit System will use a dedicated I/O control bit to periodically reset and re-synchronize the module to prevent unexpected loss of sync. The MPC3008 requires a lower clock speed due to its internal Analog-to-Digital converter electronics conversion of the input signal into a 12-bit digital value. Two analog channels are currently used with the heart rate monitor requiring 10 acquisitions per second, with temperature acquisition being converted at the same time for convenience as these parameters do not change rapidly. GPS information is input to the RP3 via a USB connection and provides NMEA standard 0183 output using a simple serial protocol. The GPS module used is an Adafruit Version 3 Ultimate Breakout, with a -165dBm input sensitivity and capable of receiving 66 channels with 10 Hz updates. It is possible to connect an external active antenna to its input to increase its sensitivity to in excess of -185dBm when the environment provides weak signal conditions. GPS coordinate information is provided to a Python program running in the background and feeding the Flask webserver each time it packages data to send to the client webpage. A hardware shutdown button is provided to insure the orderly shutdown of the Linux operating system. Operation of this switch causes an interrupt that executes the listen-for-shutdown.sh shell script. The shutdown script is available in the Smart Suit detailed documentation.

4. SMART SUIT SYSTEM

As indicated earlier two separate software modules and applications are automatically started during the system power on operation:

1.) The Lepton Thermal Imager and
2.) The Flask server.

4.1 Thermal Imager Module

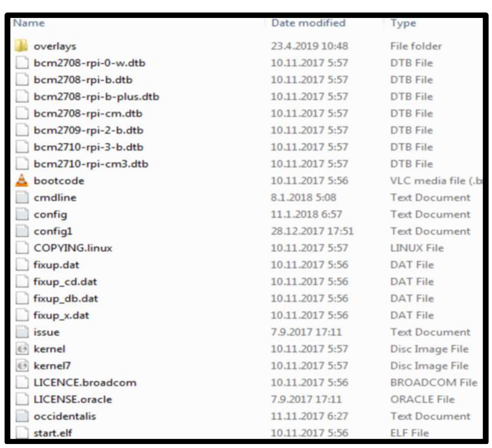
When the Lepton Thermal Imager application is started it creates a “LeptonThread” object to set up the operating parameters, such as SPI clock speed and frame size, then enters a continuous working loop. The loop inputs a line of data from the thermal camera consisting of row number plus 80 pixels, while scanning for and then synchronizing to the first line count number of a frame. Once the first row is identified the thread continues inputting lines of pixel data, storing them in an array, until the line count reaches the maximum (currently set to 60 lines), after which it scans the data array to find its max and min pixel values. Finally, it invokes an update image operation with arguments consisting of data array, with maximum and minimum values for further processing in the main thread. When the main thread receives a completed frame it pseudo colors the pixels (using a selected color map) based upon the maximum and minimum values present, to form a completed image frame. The completed image frame is written as a high-quality JPG to working memory based upon a periodic timer event, which at present is limited to 1.8 seconds per frame.

4.2 Flask Server Module

When the Flask server is started it first initializes the GPS system and senses that it is actually attached to the RP3. If the hardware is not present it keeps looking for attached hardware and will not continue until this important component part of the system is connected. When present, the GPS subsystem runs continuously in the background to populate time and coordinate data arrays for use by the webserver. When the webserver receives a request from the client application it utilizes the populated GMT time and coordinate data arrays to construct a data structure (python dictionary). Additionally, the collected physiologic data (heart rate and temperature) are appended to the data structure. When all data has been incorporated, the webpage template is rendered to contain the newly updated data, along with other java script operations to finalize the page.

5. WI-FI SSETUP AND OPERATION

Since the Smart Suit System uses a local area network to transport and to present its data, it must first be connected to the network. The Smart Suit System uses a fixed dedicated network IP address (e.g. 192.168.0.99), which will be different for each Smart Suit System, and must fall within the range of addresses used by the network configuration. Each Smart Suit System acts as its own web server and is accessed on a fixed IP address at port 5000 (such as 192.168.0.99:5000). Wi-Fi setup of the system requires that a specific file contain the Service Set Identifier “SSID” and passphrase “PSK” for the network. For a “headless” system (one without monitor, kbd, and mouse) these two items must be known ahead of time and added to the configuration file. This file may be set up prior to connecting to the Wi-Fi network for the first time, which should allow for automatic connection thereafter when the Smart Suit system boots. In a “headless” system, this configuration file must be positioned in the root directory on the RP3, which then gets automatically transferred to the **/etc/wpa_supplicant** directory upon first bootup. The easiest way to setup this file is to edit it directly on the micro-SD card using a PC. This may be accomplished by using a micro-SD adapter plugged into the PC. Note that the boot sector of the micro-SD card is readable by the PC (under Windows) because it has the correct FAT32 structure. A text editor (Notepad) may be used to edit this file (**wpa_supplicant.conf**) in the micro-SD cards root directory. One has to make sure that no non-text or other characters are inadvertently entered in the file. The detailed ontents of this file is in Smart Suit system documentation and it is used when setting up the system at some particular location with a specific WiFi Access Point. After booting, Wi-Fi configuration file will be re-located to RP3 directory: **/etc/wpa_supplicant/wpa_supplicant.conf.** If the Wi-Fi does not appear to start up during operation it could be due to an error in the original **wpa_supplicant.conf** file (such as tabs, or other wrong characters, or formating) that had been placed in the micro-SD card's /boot directory. If HDMI display output with keyboard and mouse is available for the RP3 then editing the **/etc/wpa_supplicant.conf** file directly with the correct parameters should restore Wi-Fi hardware operation. This file may contain multiple network entries to allow the RP3 Wi-Fi to automatically connect at a number of locations. Since the Smart Server System webserver delivers its webpage to a fixed (programmedin) IP address, for example **http://192.168.0.99:5000**, the router (WiFi Access Point) used must not automatically assign some other device to this fixed address value. It might be necessary to set a reserved static IP address in the router for this purpose, but one needs to check the local router set up first. When finally connected to the network, one should use a newer Edge Microsoft web browser, or an updated Firefox browser to view the webpage at the above IP address. It is not recommenbded touse Chrome on PC, as it caches the thermal image then only uses the first one received with none of the new updates being viewable. It is possible to use a smart phone or tablet web browser to see the webpage at above IP address. This may later be useful in using a tablet directly as one of the Smart Suit components, for example for some control purposes [13]. Figure 5 shows a partial list of configuration files.



Name	Date modified	Type
overlays	23.4.2019 10:48	File folder
bcm2708-rpi-0-w.dtb	10.11.2017 5:57	DTB File
bcm2708-rpi-b.dtb	10.11.2017 5:57	DTB File
bcm2708-rpi-b-plus.dtb	10.11.2017 5:57	DTB File
bcm2708-rpi-cm.dtb	10.11.2017 5:57	DTB File
bcm2709-rpi-2-b.dtb	10.11.2017 5:57	DTB File
bcm2710-rpi-3-b.dtb	10.11.2017 5:57	DTB File
bcm2710-rpi-cm3.dtb	10.11.2017 5:57	DTB File
bootcode	10.11.2017 5:56	VLC media file (.b
cmdline	8.1.2018 5:08	Text Document
config	11.1.2018 6:57	Text Document
config1	28.12.2017 17:51	Text Document
COPYING.linux	10.11.2017 5:57	LINUX File
fixup.dat	10.11.2017 5:56	DAT File
fixup_cd.dat	10.11.2017 5:56	DAT File
fixup_db.dat	10.11.2017 5:56	DAT File
fixup_x.dat	10.11.2017 5:56	DAT File
issue	7.9.2017 17:11	Text Document
kernel	10.11.2017 5:57	Disc Image File
kernel7	10.11.2017 5:57	Disc Image File
LICENCE.broadcom	10.11.2017 5:56	BROADCOM File
LICENSE.oracle	7.9.2017 17:11	ORACLE File
occidentalis	11.11.2017 6:27	Text Document
start.elf	10.11.2017 5:56	ELF File

Fig. 5. An excerpt from smart suit set of configuration files

6. IMPLEMENTATION

6.1 Components

The Smart Suit System platform components are shown in Figure 8. Their typical cost is summarized in Table 1. The total component cost is less then $150 in small quantities. These components were chosen because they were readily available and more choices exist todaya at smaller prices, so we can assume thetotal cost to be less trhan $100 in large quantities. More sensors can be added as well per customer requirement. In any case we did not optimize either the cost or the design configuration. The aim was to have a electronic demonstration anfd marketing platform as well as a general electronic design platform. The biggest challenge in the project was to design this platform in a very short period of time which was only 3 months. Additional 2 weeks were needed to implement the components into the suit itself and test it. All the components were imbedded inside the suit with addition of couple of pockets to hold the components and make them available for set up and system activation/reset. The GPS sensor was placed up in the sholder of the suit, and its antenna on the other shoulder. The thermal camera was situated in the upper pocket. The computer module with the battery is in an inner pocket secured with velcro, heart rate and temperetaure sensors were located in one of the suit sleeves. Also, cabling was embeeded throughout the suit to connect all the components. The computer module has both 4G as well as WiFi modules, and WiFi was used to connect the suit with Internet via local WiFi access point. A Micro SD card had all the drivers and software required to run the suit. The suit was tested in a number of locations around Sarajevo, Bosnia and Herzegovina and Silicon Valley, USA. For example Figure 4 shows one street in Sarajevo and Figure 3 a street in Silicon Valley, with Google Map in lower left corner indicating the street, as well as showing a person inside the building with his tempereture and heart rate data and his thermal image obtained from the camera in the suit upper pocket. This data was obtained by logging into the suit „web site“ which showed what the suit condition at that point in time. The platform is suitable for a specific sensor(s) extension as it may be required by a specific application at hand.

Table 1. Component Cost in Small Quantities

Description	Unit Cost
Raspberry Pi 3 – Model B – ARMv8 with 1G RAM	$35,00
Aluminum Heat Sink for Raspberry Pi 3 15 x 15 x 15mm	$1,95
Pi Model B+ / Pi 3 Case Base – Smoke Gray	$5,00
Adafruit Ultimate GPS Breakout – 66 channel W/10 Hz updates – Version 3	$39,99
GPS Antenna – External Active Antenna 3 – 5V 28dB 5 Meter	$12,95
SMA to uFL/u.FL/IPX/IPEX RF Adapter Cable	$3,95
USB to TTL Serial Cable – Debug / Console Cable for Raspberry Pi	$9,95
Pulse Sensor Amped	$25,00
TMP36 - Analog Temperature Sensor	$1,50

6.2 Applications

The smart suit platform as dscribed in this paper is uitable for many different applications, both for defense and for a variety of commercial applications [13] The suit in the right upper corner of Figure 6 was supplied for demonstrating purposes, and a smaller and ligher suit (jacket) can be also used. The components from Figuire 8 were physically implemented in that demo suit. The suit was demonstrated to a number of potential users such as local police, mountain rescue group, as well as civil protection service and several defense application users. Each of the potential customers indicated their own specific requirements, in particular sensor related details. Our Smart Suit Design Platform can accommodate adding additional sensors and integrating them via embedded computer and software developed for the Platform. Sensors can be either wired or wireless, and each customer may supply their own specific suit or jacket. One of the applications which emerged from potential customers following demonstrating of our design was a need to have personel positions and their health as well as kinetic (moving not moving, running, walking) information available at all times within a company or other campus, or a large building. For this WiFi would be good enough. For field applications one would need 3G or 4G for wide area wirelesss networking. Our deesign has both of these options built in.

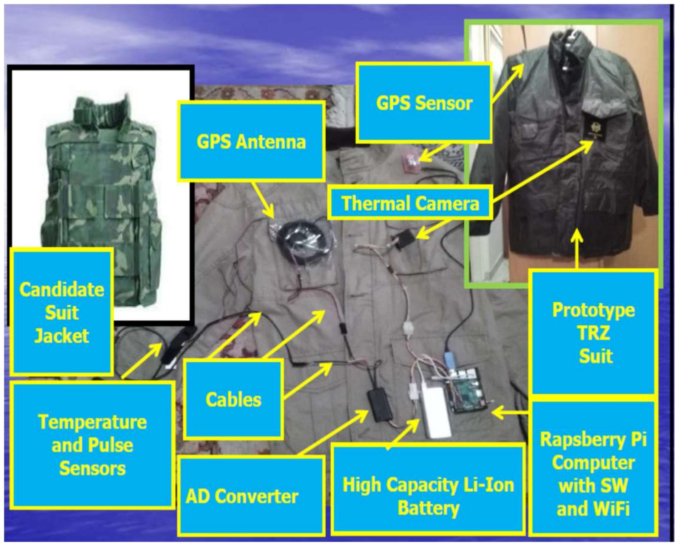


Fig. 6. Smart Suit Prototype and Components Layout

7. CONCLUSION

In this paper we present a prototype design of a smart suit (jacket) which uses GPS, temperature, heart rate, as well as thermal information sensor embedded into a specific suit for demonstration purposes. The suit has its own Internet address and as such can be accessed remotely and its condition can be observed in real time every second. Real data transfer which the platform can accommodate is much larger. Other sensors can be added as well. WiFi is also vailable for the data commnications, as well as 3G or 4G mobile communications if required by a specific application. These applications range from variety of commercial to specific defense areas.

8. REFERENCES

[1] FLIR Lepton Camera HW and SW Documentation (on line)
[2] Raspberry Pi 3 HW and SW Documentation (on line)
[3] Wearable Technologies, Wikipedia (2019)
[4] Medical and Military Vests, Contract Sewing and Wearable Device Blog (2019)
[5] "Benelux Invest $150 million in Infantry Smart Vests" available at http://defense-update.com/20150702_smart_vest.html, (26-08-2019)
[6] Accenture, Delivering public service for the future (2014)
[7] European Commission (2016), Smart Wearables: Reflection and Orientation Paper
[8] Wearable Electronics (2017), http://www.researchandmarkets.com/reports/302653/ (25-08-2019)
[9] Gordon Walace at all, Smart nanotextiles: A review of materials and applications (2007), U. of Wollongong Research Online
[10] "Smart Safety Jacket" available at https://contest.techbriefs.com/2017/entries/electronics-sensors-iot/8153 (25-08-2019)
[11] "Infrared and Thermal Imaging Systems Market by Technology (Cooled, Uncooled), by Subsector (Commercial, Military), by Application (Transportation, Security & Surveillance, Thermography, Military Vehicle Vision, Soldier Portable Vision, & Unmanned Systems), & by Geography (North America, Europe, Asia-Pacific, the Middle East, Latin America & ROW) - Forecasts & Analysis 2014–2019" available at http://www.marketsandmarkets.com/pdfdownload.asp?id=229168627(24-08-2019)
[12] M. Hodžić and I. Muhić (2014), Internet of Things: Current Technological Review, PEN, Vol. 2 No. 2
[13] Smart Suit Presentations (available from the authors) (2019)