**Characterizing a Firefighter’s Thermal Exposure in a Fire Environment**

A number of challenges arise when acquiring data from a firefighter’s thermal exposure in a fire environment. For example, the data acquisition system needs to be durable and able to endure the elevated thermal conditions encountered in fire environments. Also, the system needs to accompany a firefighter as he/she moves throughout the fire environment to conduct various tasks without affecting the firefighter’s range of motion. Additionally, cooling water must be circulated around sensors that measure heat flux, an important thermal measurement of heat rate per area.

**Overview of Portable Measurement and Data Acquisition System**

To overcome the difficulties described above, a portable measurement and data acquisition system was designed to be used with firefighter personal protective equipment (PPE). The system is composed of two main parts: the helmet and the pack.

[picture of full system]

Helmet Portion

The helmet portion of the system consists of a type K thermocouple and a Schmidt-Boelter heat flux gauge (SBG) to measure temperature and heat flux, respectively. Both sensors are mounted to the aluminum shield on the front of a firefighter helmet.

[picture of helmet, maybe pic of rubber tubing/SBG connection]

Two sections of 1/8” diameter rubber tubing used to transport the cooling water to and from the heat flux gauge connect to the SBG on the inside of the helmet. The rubber tubing and sensors’ wires travel out the backside of the helmet and to a hydration backpack.

Pack Portion

A data logger, water reservoir, and miniature water pump are contained in various pockets of the hydration backpack.

[picture of pack]

The data logger rests in the front pocket of the pack, the water reservoir used to store the cooling water for the SBG is located in the rear pocket of the pack, and the water pump used to circulate the cooling water is located in a side pocket of the pack.

[picture of FF wearing system]

The low-profile hydration pack is worn under a PPE coat on the chest of the firefighter to avoid interference with the Self-Contained Breathing Apparatus worn on the firefighter’s back. Because the hydration pack is worn underneath the PPE coat, its components are protected from the intense thermal conditions encountered in fire environments.

**Transition from Original Data Logger to Arduino Yún**

Recently, I decided to replace the original data logger with an Arduino Yún microcontroller board to expand the capabilities of the portable measurement and data acquisition system, specifically to allow for the real-time plotting of data measured by the portable system in a fire environment. To

Workflow of Original Data Logger

The data logger originally used with the portable measurement system was very basic. It was able to convert the thermocouple voltage to temperature. The converted temperature and the voltage from the heat flux gauge were received and stored by the logger every second. Only after the firefighter equipped with the portable system had exited the fire environment could the logger be retrieved from the hydration pack, connected to a computer, and have it’s data exported to a .csv file. Once the data were exported to the .csv file, a python script was used to convert the heat flux gauge voltages to heat flux and plot the temperature and heat flux data collected during the exposure. It was at this point that the data analysis process began. Workflow

* Couldn’t analyze data in real time
* Data logger is the only place data are saved until logger is connected to computer and data set is manually exported to .csv file by user
* .csv contains heat flux gauge voltages 🡪 need to be converted by python script before plotting

The workflow improves significantly when the Arduino Yún is used as the data logger. The Arduino Yún’s built-in WiFi support is used to transmit the sensor data to a host computer in real time, which is important for several reasons. First, the data are now saved in two distinct locations (the Arduino’s microSD card and the host computer) during the data logging process. Additionally, the host computer utilizes the Bokeh server to plot each datum point as it’s received in real time. Finally, the workflow has been simplified: the last section in the original workflow is no longer required when the Arduino Yún is used as the data logger because the data file and plots are generated and saved on the host computer as data are logged.

**Basic Set Up and Initialization before Data can be Plotted in Real Time**

An ADC breakout board was added to the microcontroller board, and the Yún was configured to use the breakout board to achieve desired resolution during analog-to-digital conversion of the sensors’ voltage signals.

The Arduino Yún supports a Linux distribution (OpenWrt-Yun) and also has built-in WiFi support.

A Python script was written and uploaded to the Yún. The script uses the pika module and the Yún’s WiFi capability to connect to the message broker (RabbitMQ) on the host computer. Once connected, the voltages from the temperature and heat flux sensors are read and converted to their corresponding values. Finally, a message containing a timestamp, the time step corresponding to the current exposure, heat flux, and temperature is created. The message is sent to the broker on the host computer and is also written to a local .csv file saved on the Arduino’s microSD card.

The receive\_helmet\_data.py script running on the host computer uses the pika module to connect to the message broker. Once connected, it receives the messages sent to broker by the Arduino and adds them to a .csv file saved on the host computer.

Methodology

The plot\_helmet\_data.py script generates two plots – one for the temperature and one for the heat flux – using bokeh and the data from the helmet\_data\_output.csv file, or the same file that is updated with each message sent to the broker by the Arduino Yún. The two plots are sent to the bokeh server, which publishes them online. After the heat flux and temperature plots are generated and published by the server, the latest heat flux and temperature data sets from the helmet\_data\_output.csv file are sent to the bokeh server approximately every second, and the heat flux and temperature plots are updated with any new measurements.