IRHUB

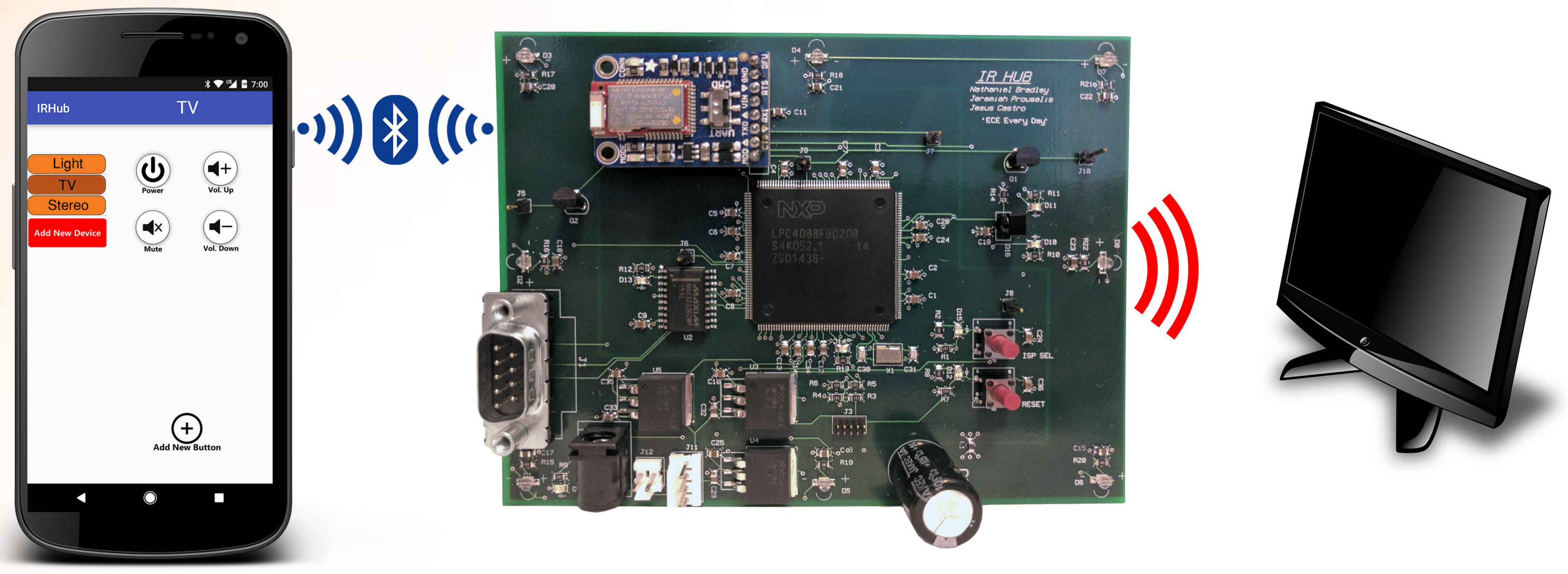
Jeremiah Prousalis • Jesus Castro • Nathaniel Bradley

What it does:

IRHub turns your phone into a universal remote using an *Infrared Receiver* to *learn* codes from your remotes, *IR LEDs* to *transmit* those codes, and a *Bluetooth Low Energy* connection to an *Android Application* that puts you in *control* of the Hub.

Android Application

- *Organize Devices and Buttons*
  - Buttons are grouped by device tabs
  - Users may add remote buttons they wish to control from the app
  - Remote signals are *not* stored on the App
  - Instead, buttons are assigned a unique ID
  - Adding button sends unique ID to the Hub
  - Pressing a button sends unique button ID to the Hub for transmission



940nm Wavelength IR LEDs

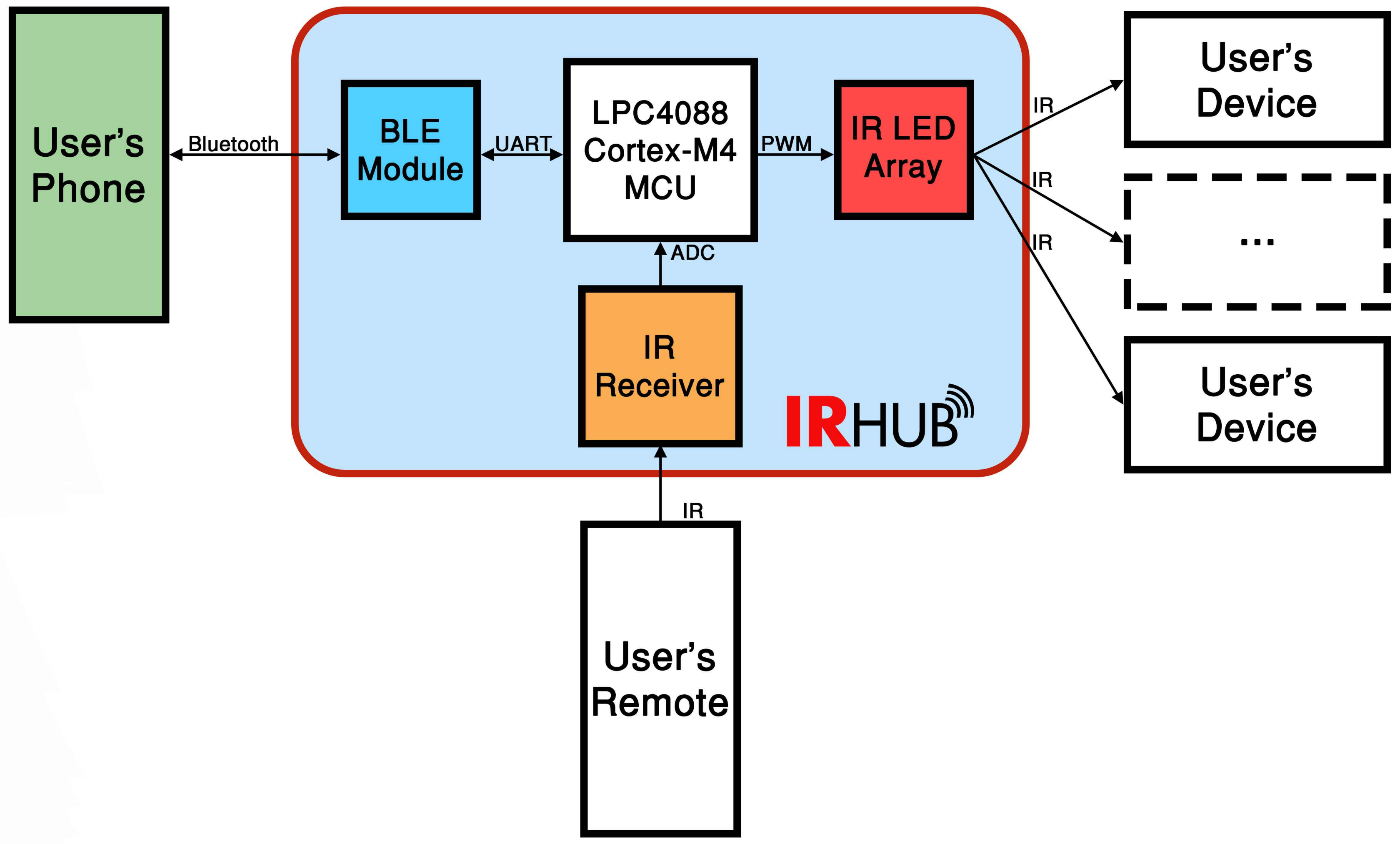
- During Transmit, PWM from the MCU drives emitter array
- Code indexed by <ID> is recreated
- Button signal is recreated based on stored code
- 8 IR LEDs around perimeter of the board broadcast the code providing 360° room coverage
- Positioning the Hub in center of room allows signals to reach and control any devices within line of sight of the board

Adafruit Bluefruit LE UART Friend

- Nordic UART connection profile acts as transparent data pipe between Android's Bluetooth connection and UART on the Hub's MCU

Nordic UART Service

- *TX Characteristic*
  - Phone can read Hub state feedback via this characteristic
- *RX Characteristic*
  - Phone can send two commands via this characteristic
    - "R"+<ID>: Read signal & store code at index of ID
    - "T"+<ID>: Transmit code stored at index of ID



940nm IR Photodiode

- Receiver for learning remote codes
- During "Learn" state, the MCU waits for input on the 12-bit ADC
  - User points their remote at the Hub and presses the button they want it to learn
  - Signal edge triggers ADC sample at 200kHz
  - Signal is decoded, compressed, and stored in on-board EEPROM at location determined by <ID> provided by phone
  - Up to 30 button codes may be stored at one time

UCSB CE Capstone
Fall 2016 / Spring 2017