Wireless Camera

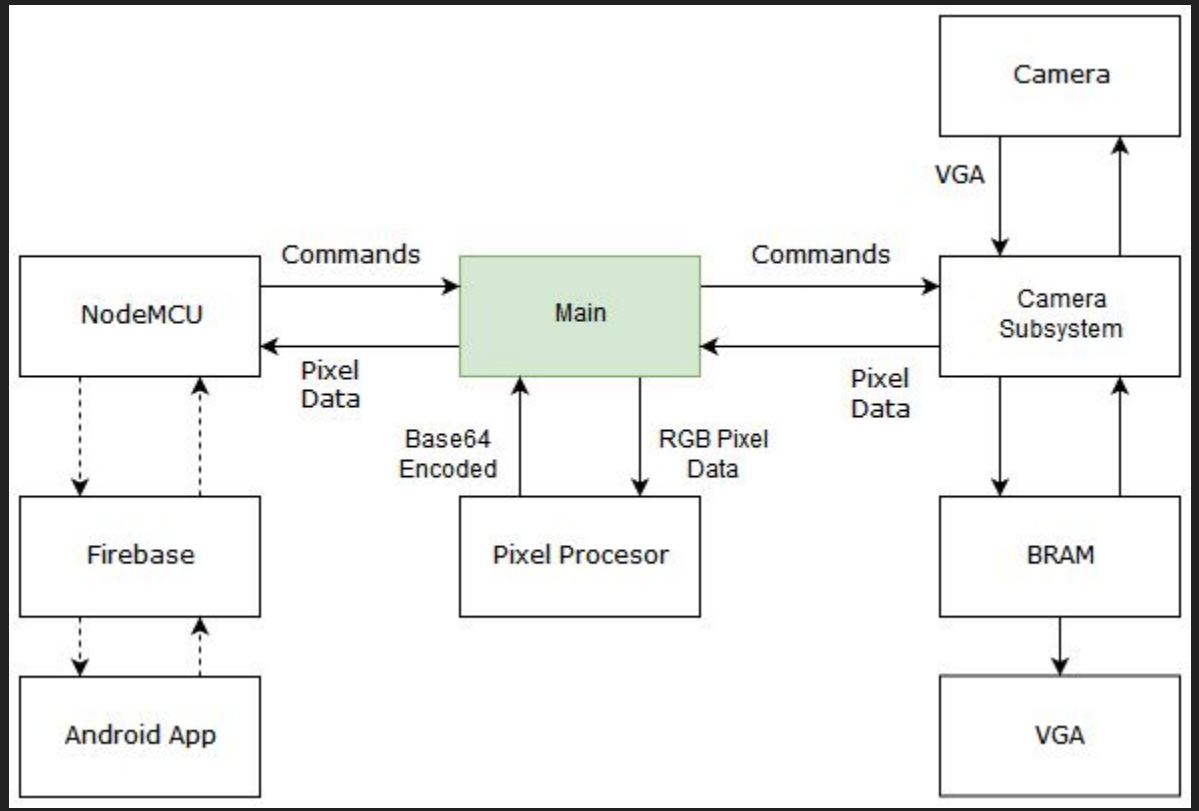
Wireless Camera using Android/Firebase and FPGA Camera

By: Andrew Capatina, Ryan Bentz, Ryan Bornhorst

Team Roles

- Andrew Capatina
  - Android App
- Ryan Bentz
  - Image Processing
  - Embedded Software
  - Wifi Connectivity
- Ryan Bornhorst
  - Configure Camera Display
  - Image Processing

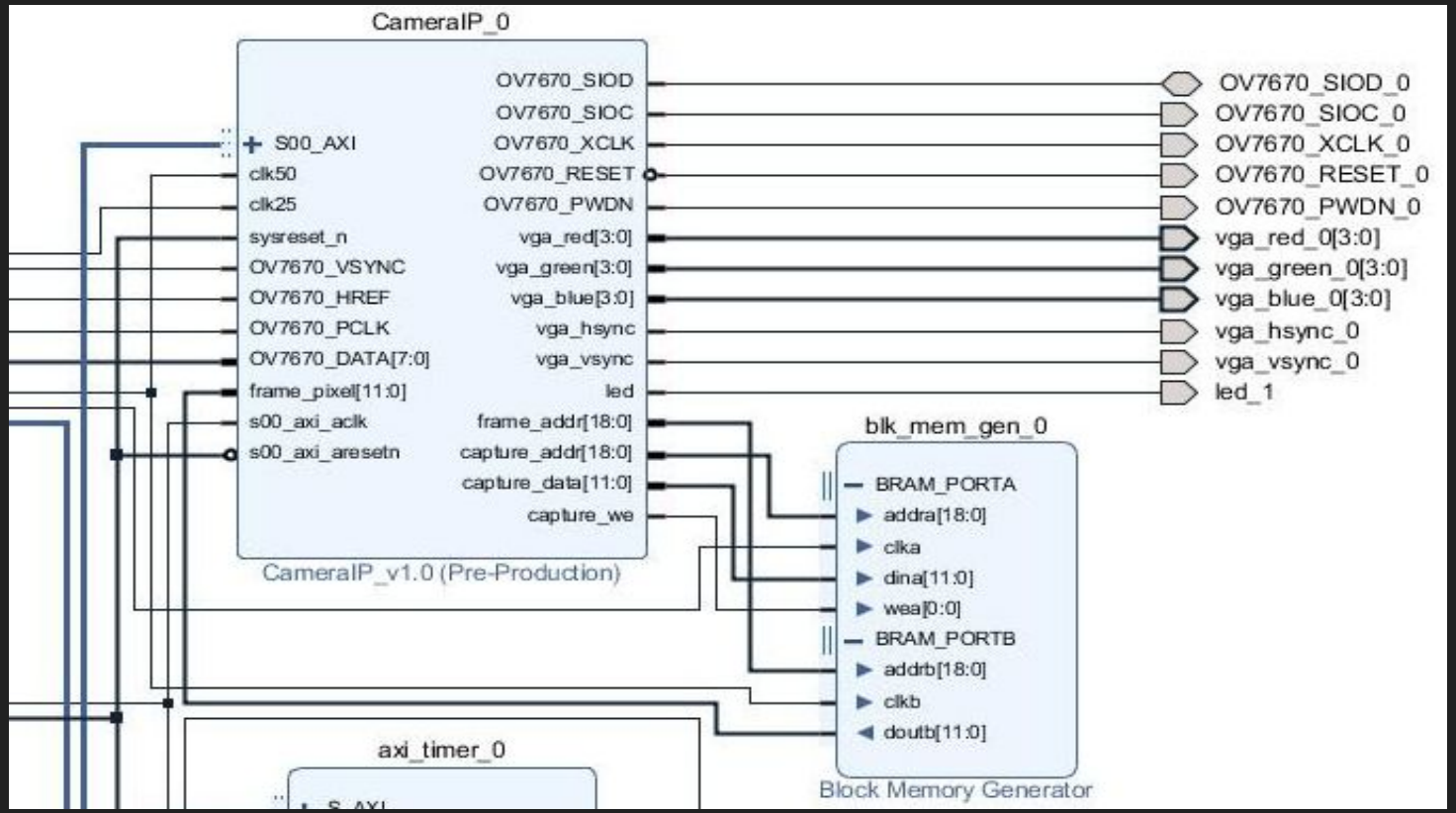
System Design



OV7670 Camera Module

- Controlled through Serial Camera Control Bus (SCCB)
- Supports multiple formats - we are using RGB565
- Max image transfer rate - 30 fps / VGA
- Use VGA display to debug image quality

Camera Module as IP



Camera Status Report

Completed:

- Camera setup as IP that can be read/written to from software
- Streaming the picture through the VGA port
- Using I2C bus to write to camera for configuring display output
- Able to freeze the camera frame
- Able to capture the image array during a frozen frame through the AXI bus using software

Snapping a Picture

- Due to memory limitations and using the UART we scaled the image size down to 160x120
- Freeze Frame signal in Camera IP
  - Usually the RAM write address is auto-incrementing to continuously refresh the image from the camera.
  - To take a picture, freeze signal goes high, multiplex the address line to the AXI bus so that we can manually read the already captured data from the BRAM.
  - Image address is written onto the bus and pixel data is read off the bus to be transferred to the NodeMCU.
- After freeze frame signal goes low, read/write address control goes back to the camera.

Challenges Faced

- Original image -> 640 x 480… scaling it down to 160 x 120 required rewriting the hdl for capturing the image.
- Not enough BRAM to have more than one image stored. Required multiplexing the address line to the BRAM for VGA display/image capture.
- Had to reference some other projects for adjusting the settings to get a viable image to display. Our project involved setting ~76 camera registers for the display settings.

Embedded System

- Standalone OS
- Main loop waits for input from Android app or button press
- Transfer Process

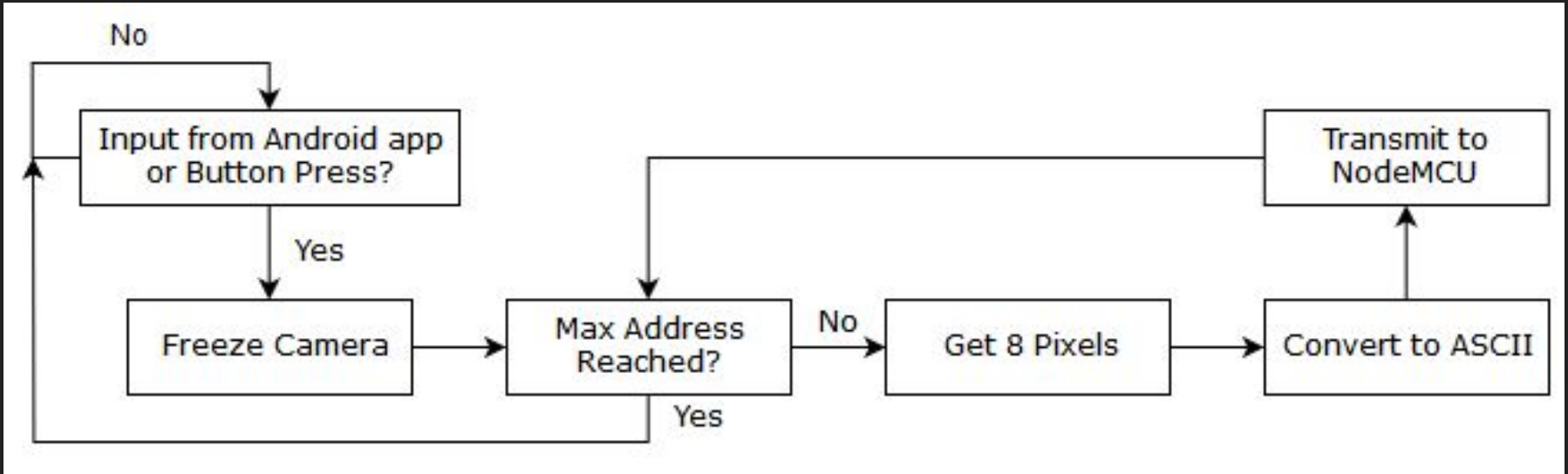


Image Transfer

Base64 Encoding

- Binary pixel data encoded before transmitting to Firebase
- Firebase can't handle raw binary data

Algorithm

- Source binary data split into groups of 6 bits each
- 6-bit value is used as index to an array of 64 alphanumeric chars
- Different types of encoding schemes (some expect a new line inserted every 76 characters)

Image Transfer

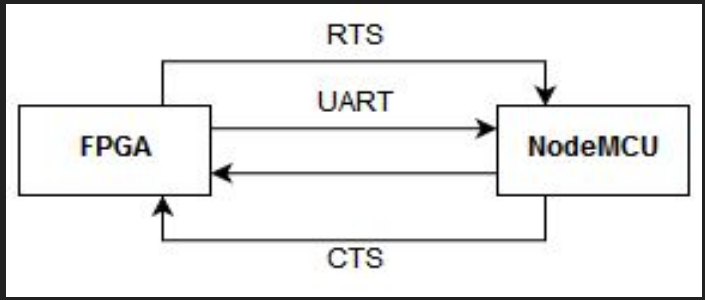
Android Image Decoding

- Android app listens for new strings and concatenates them together
- String converted to array of chars which is decomposed into RGB values
- Bitmap created in Android using Bitmap Methods
- Bitmap stored to Firebase Storage

Communication

FPGA - to - NodeMCU:

- Uses a half-interlocked communication scheme
  - NodeMCU toggles CTS pin when it is OK to send data
  - LOW = OK to send, HIGH = Wait
- UART data sent in 16 byte bursts
- NodeMCU collects the data into a 1K array of char



Communication

NodeMCU - to - Firebase:

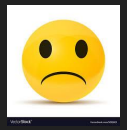
- Base64 data stored as a 1K array of char
- Data packaged in JSON Variant for transfer
- NodeMCU sends signals to Android app when transfers begin and end
- NodeMCU periodically checks database for changes from application to take a picture

Communication

NodeMCU

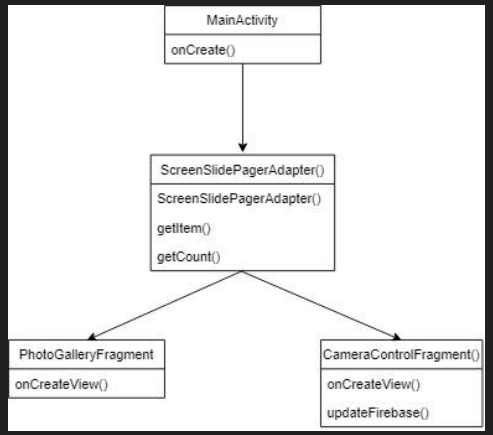
- Infinite loop listens for incoming data from UART
- Timer periodically checks Firebase for commands from Android application
- Commands from Android app are sent via UART
- Program monitors transfers to Firebase and halts transfers if Firebase error occurs

Challenges Faced

- FPGA to NodeMCU communication
- NodeMCU functionality (Arduino )
- Uploading image data to Firebase
- Decoding image data in Android and recreating the image
- Vivado SDK

Android App

- What has been completed:
  - User interface for camera interaction.
    - Firebase is updated once user makes selections.
  - Displaying lists of images.
    - A fragment receives an image ID each time it’s instantiated.
- What is remaining:
  - Splash screen with Async task.
  - Ability to delete photos.
  - Listeners for adding new images to layout.
- Problems encountered:
  - Implementing RecyclerView for displaying a gallery of photos.
  - Performance issues when using Firebase.



DEMO

THANK YOU