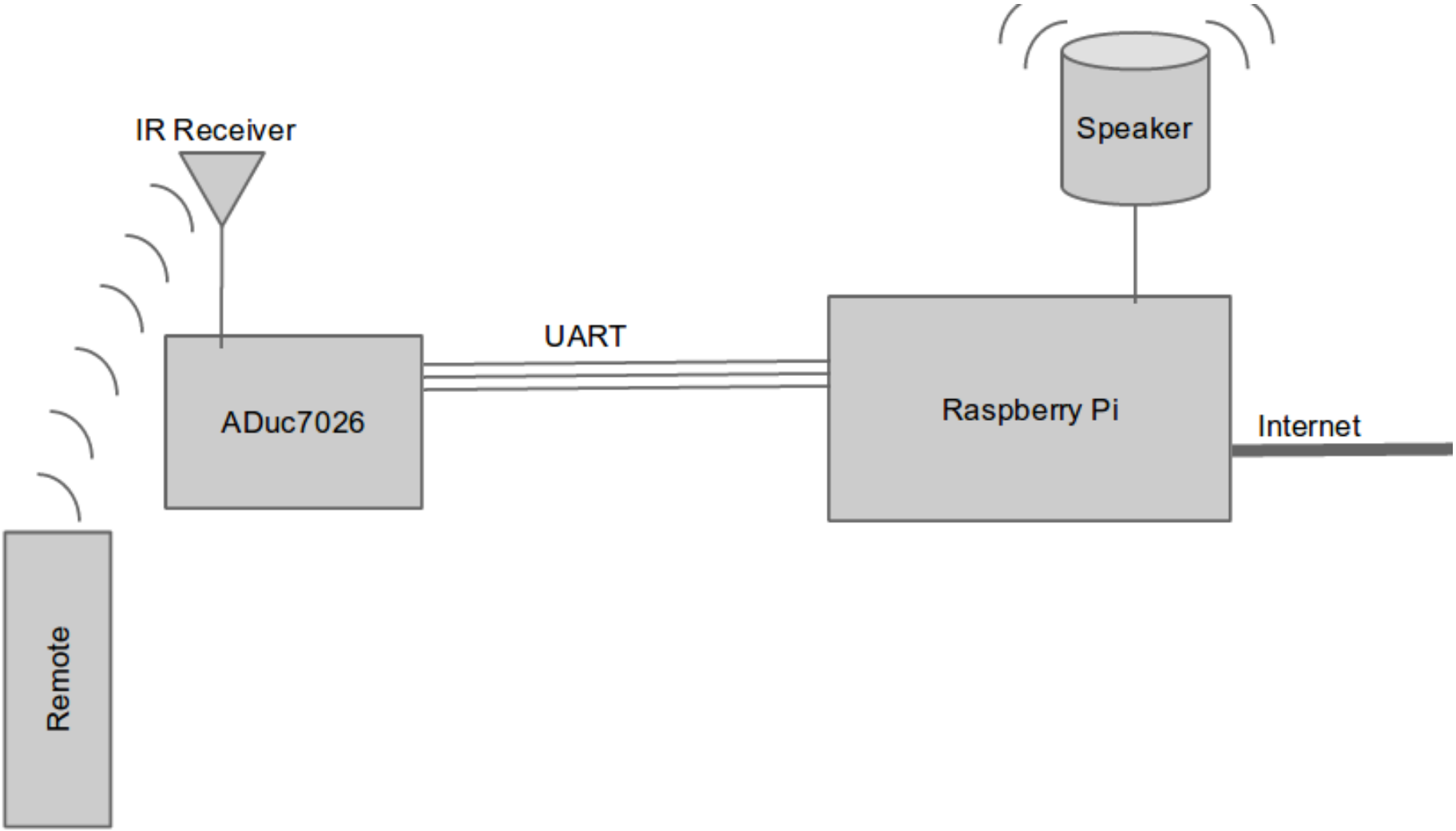
A Remote-Controlled Pandora Radio Player

Jason Chaves

AP 208 Project

6/3/14

Block Diagram



Division of Labor

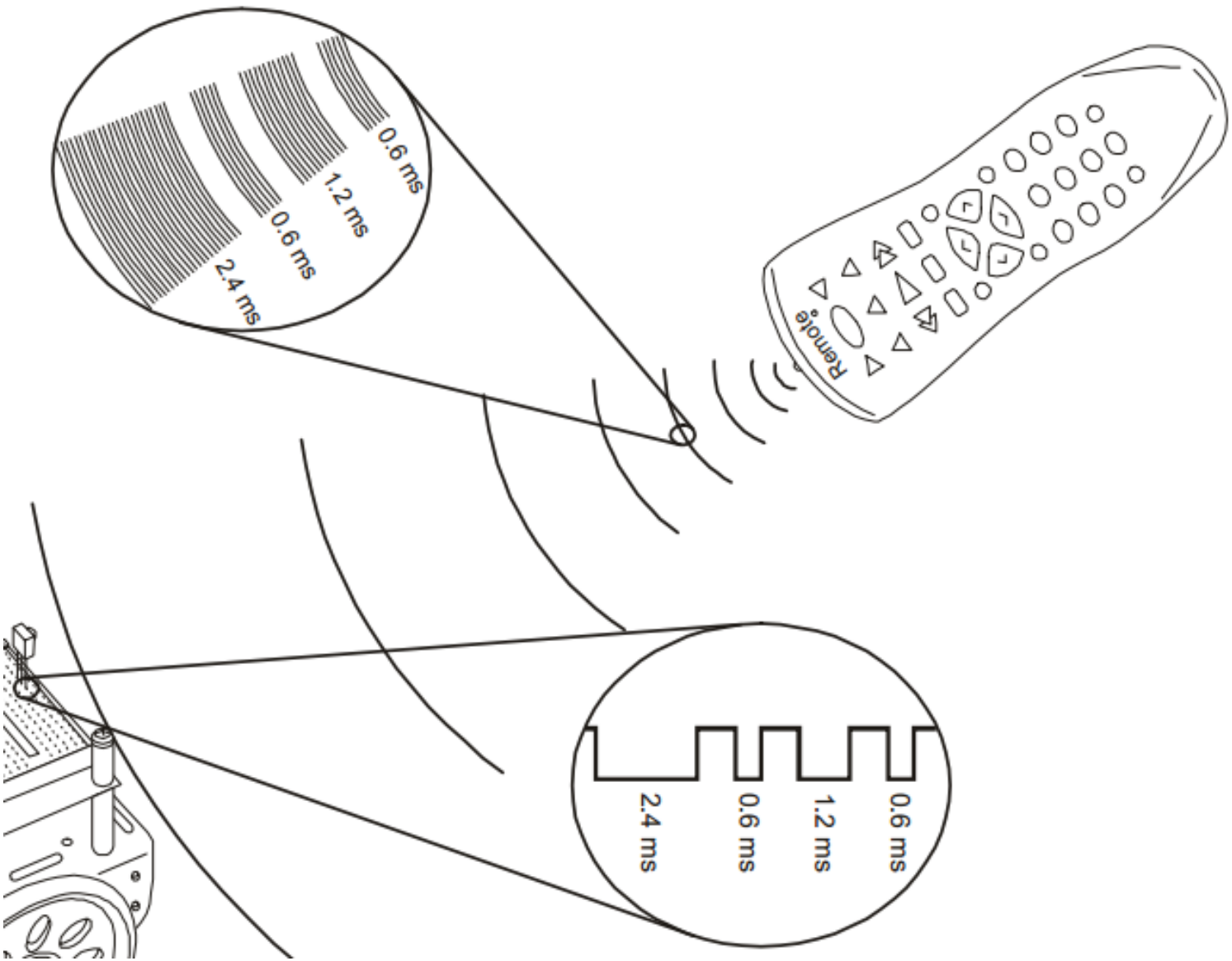
Micro-controller

- Read in IR Receiver signal on GPIO
- Decode remote message
- Send characters to Raspberry Pi

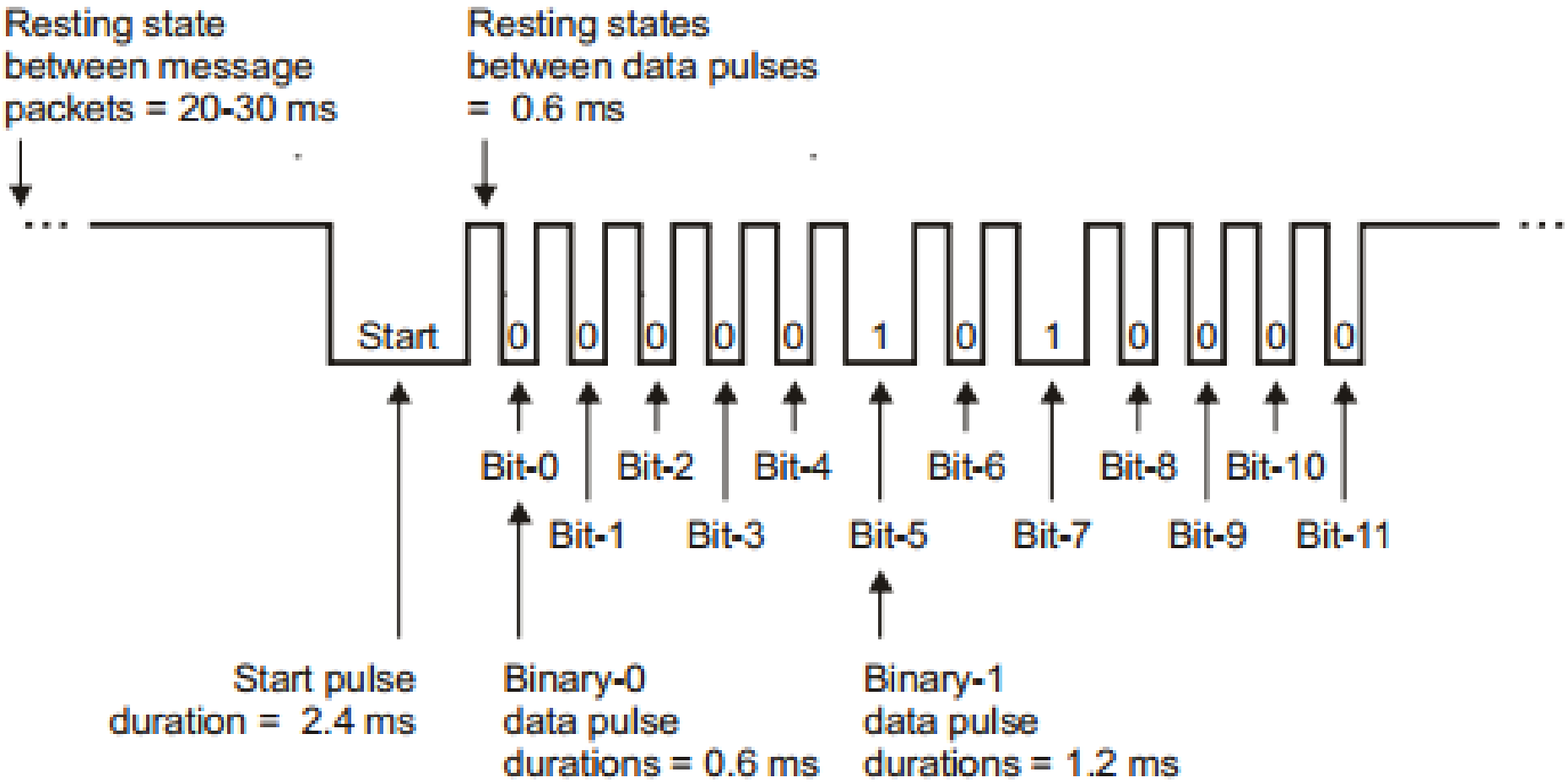
Raspberry Pi

- Receive characters from micro-controller
- Interpret character/command
- Communicate with Pandora Radio client program

Remote IR Messages



Remote IR Messages



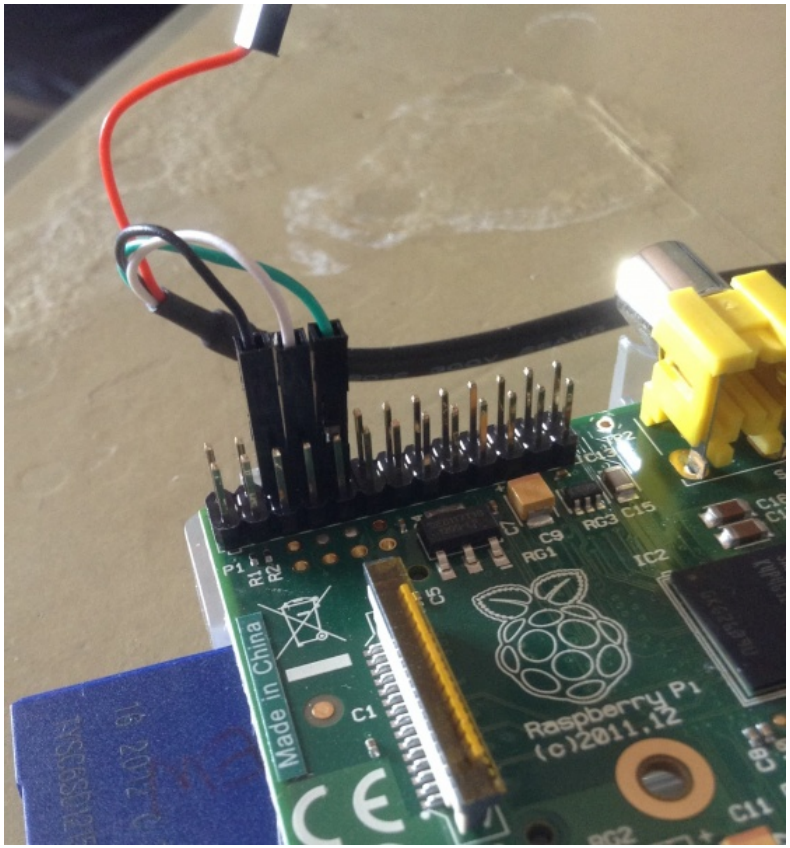
Measuring a Single Pulse

- Used on-board Timer1 to measure the duration of a pulse
- Used on-board PLA to invert the negative-true IR Receiver signal
- The negated signal was used as to trigger a PLA Interrupt, to transfer into interrupt handler code
- The negated signal was AND-ed with the 45 MHz core clock, to produce the clock for Timer1
- So Timer1 only counted when the IR Receiver signal was asserted, and it always started from 0
- Interrupt handler code waits until the interrupt (IR signal) stops being asserted, reads Timer1 value, converts to ms, and compares to 2.4 ms, 1.2 ms, and 0.6 ms valid durations

Reading a Whole Message

- 16-bit short acting as bit vector for remote message
- Message is only valid if the start bit was detected
- Detecting start bit clears the bit vector, resets the bit vector index, and declares the following message valid
- Bits OR-ed into short in LSB-first order
- After 11 message bits have been read, bit vector divided into button variable and mode variable
- If the mode correctly corresponds to the device, translate the button variable into a character to send to RPi over UART
- Each button press on the remote sends multiple messages (20-30 ms apart), so after one is read, wait a few ms to avoid reading other messages from the same button press

UART on Raspberry Pi



- Only RxD, TxD, and GND wires used for UART (3.3V)
- Have to use rpi-serial-console script to enable use of UART pins
- pySerial library used + Python script to read single characters from UART

Pandora on Raspberry Pi

pianobar

pianobar is a free/open-source, console-based client for the personalized online radio Pandora.

```
        28)     Shara Worden Radio
        29)     Spybreak! Radio
        30) q   Tell It Like It Is Radio
        31) q   Temoignage Radio
        32)     The Chemical Brothers Radio
        33)     To Pluto's Moon Radio
        34)     Trance Radio
        35) q   Tri-Me Radio
        36)     Trifonic Radio
        37)     Wallis Bird Radio
        38) q   Winter Radio
        39) q   Worrisome Heart Radio
        40) q   You Don't Know My Name Radio
[?] Select station: 3
|>  Station Breaks Radio
(i) Receiving new playlist... Ok.
|>  Manmade Symphony by Aquasky on Bodyshock
         0) Uberzone - Seville (General Midi Mix)
         1) Quantic - Life In The Rain  <3
         2) Hi-Bias - Drive It Home
|>  Seville (General Midi Mix) by Uberzone on Y4K
         0) Quantic - Life In The Rain  <3
         1) Hi-Bias - Drive It Home
|>  Life In The Rain by Quantic on The 5th Exotic <3
|>  Drive It Home by Hi-Bias on John Digweed: A Collection Of Classics
(i) Receiving new playlist... Ok.
|>  Old John [Future Funk Squad's 'jonboy' Remix] by Bomb The Bass on Future Chaos
|>  Get A Move On by Mr. Scruff on Keep It Unreal
         0) Yoshida Brothers - Inside The Sun Remix
         1) Propellerheads - Take California
|>  Inside The Sun Remix by Yoshida Brothers on Best Of Yoshida Brothers | Tsugaru Shimasen
#   -02:33/04:12
```

Pandora on Raspberry Pi

In a Python script, I can create a background sub-process running Pianobar.

I can then write to that sub-process's STDIN to send commands to the program

And I can read from it's STDOUT to read it's output (such as channel/station listing)

Text-to-Speech

I also have a text-to-speech script that uses Google Translate's TTS

This is used to say greetings when Pandora is turned on/off, and to read the station directory to the user.

It's called from within the master Python script

Python Script

- Python script on Raspberry Pi continually tries to read a character from the UART.
- When it reads a character, if it's not a digit, it translates it to a command to be written to Pianobar's STDIN.
- If it is a digit, it appends it to a channel number entry.
- The channel change command is sent with the populated channel number entry when the Enter button is read.

Supported Functions

- Turn Pandora Radio On/Off (Power Button)
- Play/Pause (Mute Button)
- Increase/Decrease Volum (Volume Buttons)
- Skip Song (Channel Up Button)
- Dislike Song (Channel Down Button)
- Enter Station Number (Number & Enter Buttons)
- Clear Station Number (Prev. Channel Button)
- Read Station Directory (Channel Number 00)

Existing Issues

If the remote is more than 10 feet away from the receiver, some button message can become corrupted. E.g. Volume Up $->$ Channel Up, Mute $->$ 5

To fix this, one should require that the measured durations of the bits are adequately close to 2.4 ms, 1.2 ms, 0.6 ms, to avoid mis-registering corrupted bits. A message should be invalidated if any bit has an incorrect duration.

One should also measure the time between bits, requiring it to be adequately close to 0.6 ms, to avoid splicing messages together.