Electronics Design Laboratory
Lecture #11
Project

• Must rely on fully functional Lab 2-4 circuits, Lab 5 circuit is optional
  – Can re-do wireless or replace it with a different control method if desired. The project does not have to include remote control.

• Must include an additional hardware component. Programming only add-ons or enhancements do not count.

• When selecting components check
  – Range
  – Voltage
  – Communication
Sensors

- Range
  - Ultrasonic
  - Infrared
  - Buttons
- Position
  - Compass
  - Accelerometer
  - Gyros
  - GPS
  - Tilt
- Sound
  - Voice Recognition
  - Microphone
  - Speakers
- Environment
  - Temperature
  - Humidity
  - Altitude
  - Pressure
  - Light (LDR, etc)
- Physical
  - Pressure
  - Flex
  - Vibration
  - Buttons
- Other
  - RFID
  - ZigBee
  - Bluetooth
  - Other wireless

Actuators

- Your robot
- Motors
  - DC (Brushed or Brushless)
  - Stepper
  - Servo
  - Linear
  - Solenoid
- LEDs
  - Single Color
  - RGB w/ PWM control
  - Infrared
  - 7 segment

Arduino

- Shields
  - Motor/Stepper/Servo
  - GPS
  - Audio (wave, mp3, etc)
  - Ethernet
  - GSM
  - Wifi
- Larger controllers
Final Project and Experiment 5

• Final Project
  – Project ideas and discussions this week, additional parts you may need?
  – Parts can take 1-1.5 weeks to get here. Order early and order extras!

• Experiment 5
  – Lab 5A: build the components of a wireless on/off and speed control circuit (should be done Tuesday this week)
  – Lab 5B: prelab due on Thursday
    • Write code for the Arduino to measure the on time of a digital input for speed control (should start this Thursday, finish Tuesday next week)
    • Demonstrate wireless control of the robot, demo is Thursday next week
Wireless Transmitter/Receiver

- Carrier frequency $f_c$ is fixed at 434MHz
- Modulation frequency $f_m$ is much lower
- By filtering $v_{rx}$ the sent data can be re-created
AM Modulation of RF signal at 434 MHz

Time Domain

Frequency Domain

Sinusoidal Modulation

ON/OFF Modulation

Carrier

Modulation

Signal

Amplitude Shift Keying (ASK)

0 fm  3fm  5fm  fc
RF Communication

• Group A (Red Waveforms) wants to send a 1sec pulse starting at 100ms. Group A will use a $f_m = 500\text{Hz}$ modulation frequency

• Group B (Black Waveforms) wants to send a 1sec pulse as well, but starting at 500ms. Group B will use a $f_m = 1200\text{Hz}$ modulation frequency

• Both groups are going to try and send this signal wirelessly, using a 434MHz wireless transmitter/receiver pair

This is just an example of data sent by two groups
Both signals are modulated using a 555 timer oscillating at 500Hz for Group A, and 1.2kHz for Group B.
The RF transmitter modulates the signal a second time at the carrier frequency of 434 MHz.
• Both RF transmitters are sending on the same frequency
• In addition, there is a noticeable amount of noise in most environments
• The result is an extremely messy signal, and this is with just two groups transmitting simple data pulses.
RF Communication

- When neither transmitter is operating, the RF spectrum is dominated by noise.
- This random noise floor is generally of a low magnitude and equally distributed across all frequencies.

Noise spectrum when no one is transmitting
When Group A begins to transmit, spikes in the frequency spectrum appear.
These spikes are much larger than the noise floor and at known frequencies!
• When Group B begins to transmit, additional spikes in the frequency spectrum appear.

• These spikes are at a different frequency than Group A’s transmissions!

• Neither signal overlaps with the other.
An RF receiver acts as a demodulator.
This demodulation shifts the frequency spectrum such that $f_c = 434$MHz is shifted to 0Hz.
Both Group A and Group B receivers are picking up everything.
RF Communication

- Each group’s filter is tuned to their specific modulation frequency.
- This is how we separate our signal from noise and other groups.

---

**Circuit Diagram**

- Input Signal → Modulator → Transmitter
- Receiver → Filter → Peak Detector → Robot (Arduino)

**Filter Configuration**

- Group A Filtered Output
- Group B Filtered Output

**Graphs**

- Intensity vs. Time
- Magnitude vs. Frequency

- Components:
  - R1, R2, R3
  - 10nF capacitors
  - 47kΩ resistors
  - 0.1μF capacitor
  - 5VDC power supply
RF Communication

- Each group's filter is tuned to their specific modulation frequency.
- This is how we separate our signal from noise and other groups.

![RF Communication Diagram](image-url)

![Graphs of Magnitude and Volts Over Time](image-url-1)

**Group A Filtered Output**

**Group B Filtered Output**
RF Communication

Each group's filter is tuned to their specific modulation frequency.
This is how we separate our signal from noise and other groups.
RF Communication

- Peak detector follows the peak value of the waveform
- This value is a diode drop less than the real peak!
- We can use a comparator on this peak detector output in order to generate our pulse outputs.
Complete Receiver

**Input Signal** → **Modulator** → **Transmitter** → **Receiver** → **Filter** → **Peak Detector** → **Robot (Arduino)**

**Filter** should have a gain of 1 (0dB) at $f_m$
When you are transmitting, $v_f$ should be a 0-5V sinewave centered around 2.5 V

**Peak Detector**
RC time constant in the peak detector should be large enough so that ripple in $v_{pd}$ is small.
The peak detector output $v_{pd}$ should be around 4 V when you are transmitting.

**Comparator**
Comparator $V_{threshold}$ should be close to but below the $(v_{pd})_{min}$ when you are transmitting

0-5 V digital data to Arduino

---

ECEN 2270
Electronics Design Laboratory

19
### B.3 Transmission $t_{on}$ Measurement

- **We will be using pulse length to control your robot.**
  - This is one of the simplest ways to communicate using 1’s and 0’s, but it doesn’t give you much flexibility. It is a good first step.
- **For this method of control**
  - Your transmitter input will be a string of pulses at a fixed frequency with a certain $t_{on}$
  - Your receiver will filter and re-digitize these pulses and output them to your Arduino
  - The Arduino will measure the $t_{on}$ of the incoming pulses, and take action based on the length.

You may use another 555 (or another Arduino) on the transmitter side to generate the DATA signal