

## SECTION SIX

### REMOTELY OPERATED VEHICLE v2.0

## SECTION OVERVIEW

- Use your knowledge of computer architecture to create a real system as a proof of concept for a possible consumer product
- Improve your knowledge of interrupts and ability to use them
- Learn how to use the Universal Synchronous Asynchronous Receive Transmit (USART) on the mega128

## PRELAB

1. In this project you are going to be given a set of functions that you need to have your ‘toy’ perform. Find another toy that uses a microcontroller and describe the functions that it performs. For each function, explain the interesting code (mega128 specific – setup registers ext.), mechanics, and/or electronics needed.

## PROCEDURE

### Problem

You work for TekToy Co., a global subsidiary of TekBots International. You are working on designing a new toy for the ‘remotely controlled’ line of products. You want to build a simple robot that can be controlled using IR. You want to use IR because it is cheaper and requires less design than an equivalent RF system.

Your first step is to build a proof of concept model so that you can sell the idea to management. You dig through your desk and come across your TekBots base from when you were in school and a couple of AVR microcontroller boards for the TekBot. Since you know how to use these parts you decide to build your proof of concept model with them.

### Specifications

Below is a list of things to consider while creating your design.

1. You would like to have multiple robots running at the same time so you need to somehow make a robot respond only to its own remote control. You decide to do this by assigning each robot a distinct address. The remote should transmit this address each time a command is transmitted. You have decided to send two eight bit codes, one address and one data. Address code must start with a ‘0’ and data codes must start with a ‘1.’ Figure 1 shows an example.

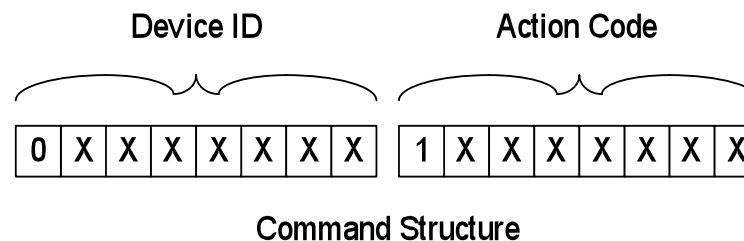


Figure 1: Example command structure

2. The AVR boards you are using have a built in IR transmitter and receiver connected to one of the serial UARTs. You feel that the UART is a good option for communicating to the robot. You will need to setup the UART on both the robot and the remote.

The AVR can set up the UART to communicate at high speeds, but you know that the IR transmitter and receiver cannot work this fast. You should use a baud rate of 2400.

3. For your simple proof of concept model, you only need a few actions: forward, stop, reverse, left, and right. To make your life easier you decide to spec out what the different commands are before you start coding. These commands are shown below in Table 1.

Command	Action Code
Forward	0b10110000
Backward	0b10000000
Turn Right	0b10100000
Turn Left	0b10010000
Halt	0b11001000
Future Use	0b11111000

Table 1: Action codes for proof of concept model

4. You can take advantage of a built in 'loop back' for the IR system on the AVR boards. When something is transmitted using the IR, the receiver can see this signal and will receive the same data. This means you can compare what you sent with what you receive to see if there has been an error in transmission. This error could be because there are multiple remotes running at one or a possible programming error.
5. You would like to have some minimal intelligence on the robot. When either of the front whiskers is triggered, the robot should reverse and turn away from the point of contact, ignoring any input from the remote control. After the robot has avoided the object it should resume what it was doing before the impact.

There is a piece of sample skeleton code available on the web. If you would like to use it go ahead. If not, feel free to do your own design.

## PART 2: FREEZE TAG

Since you are going to be showing your proof of concept to upper management, you feel you need to have something that can generate some ‘real excitement.’ The idea of having two robots play freeze tag with each other sounds like a good idea.

You think the best way to do this is to have each robot emit a certain code when the user with the remote presses a certain button. This code would then be detected by any robot that could see it (except the one ‘firing’) and cause the robot to ‘freeze’ for five seconds. When the robot unfreezes it should immediately resume what it was doing. After three freezes, the robot should stop working until it is reset.

The code your robot should transmit to freeze other robots is shown in Figure 2. The robot—not the remote—transmits this code. Make one of the buttons on the remote transmit a code to the robot that has the robot transmit the freeze code. The freeze command code from your remote should be 0b11111000.

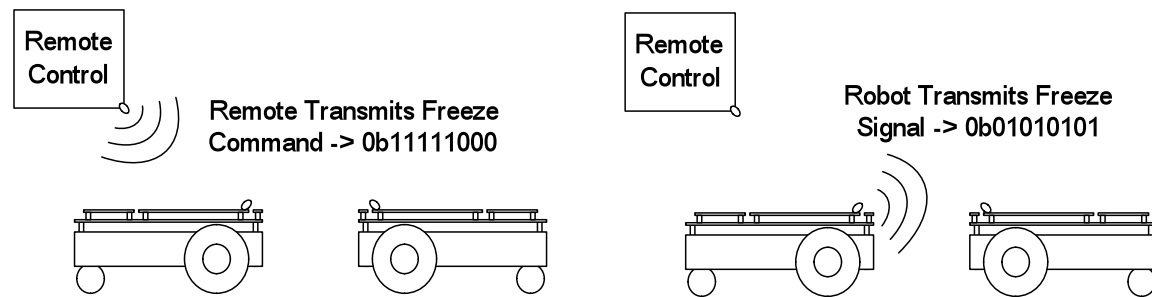


Figure 2: Freeze Tag Operation

To show your design you will need to work with another team to have the two robots compete. Show your TA and have each team turn in a copy of their code with a written description.

TA Signature: \_\_\_\_\_

## STUDY QUESTIONS/ REPORT

Write up a short summary that details what you did and why, explains any problems you may have encountered, and answer the questions below. Be sure to explain the operation of your code in the document. Do not rely only on the comments in your code to describe the function of the code. NO LATE WORK IS ACCEPTED.