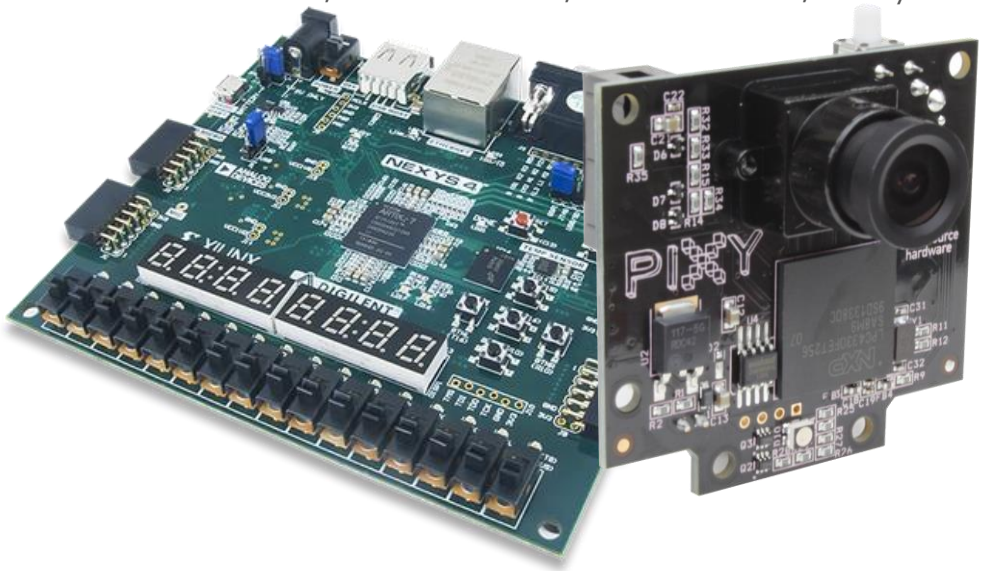
Interfacing FPGA with PixyCAM via UART Protocol

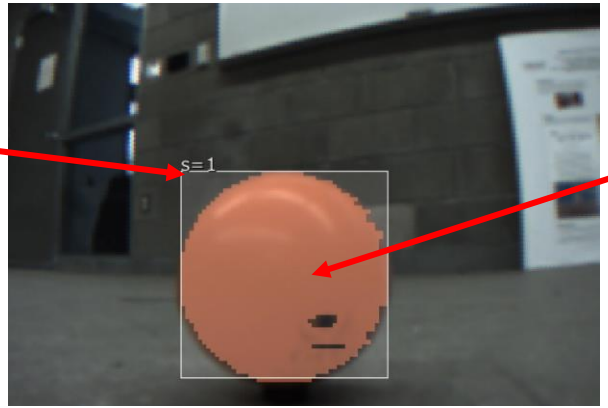
(Universal Asynchronous Receive Transmit)

Kristof von Czarnowski, Matthew Wener, Luke Pridemore, Randy Wittorp



Pixy CAM

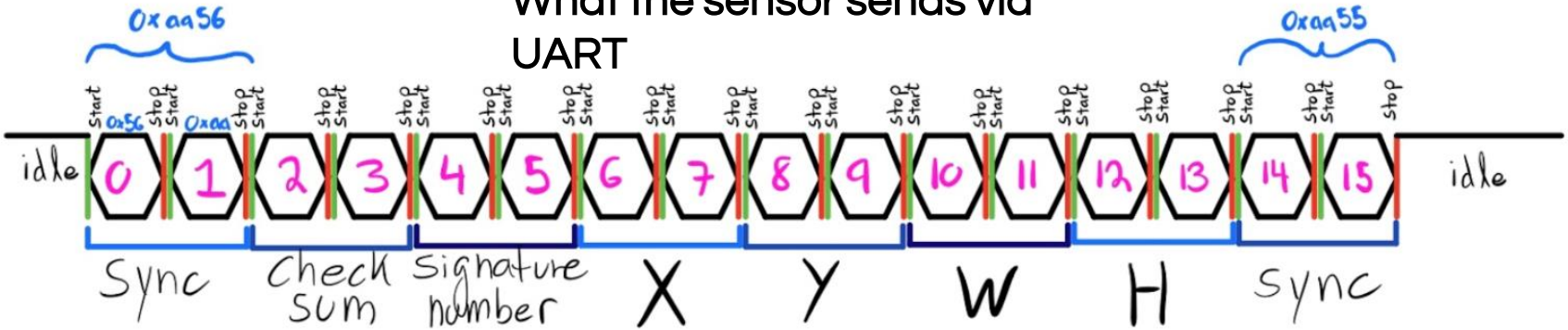
What the sensor sees.



Algorithm puts a map around the object. Calculated center point (x,y) and the dimensions.

Sensor is calibrated to seek color codes.

What the sensor sends via UART



PROJECT STATEMENT

PixyXFPGA

Goals of Project:

- ➢ Interface PixyCAM with FPGA
  - Parse a UART Signal
    - Gather all 16-bit words from protocol
  - Store information in registers.
  - Convert the information to base 2.
  - Display live information on 7-seg display
    - Display information (X, Y, W, or H) chosen using on-board switches.
    - Indicate the variable that is being displayed with an RGB LED.

Motivation:

1. To learn more about asynchronous serial communication.

2. An interesting and useful application-based project using a unique sensor.

3. Team member was acquainted with sensors capabilities and calibration.

Applications:

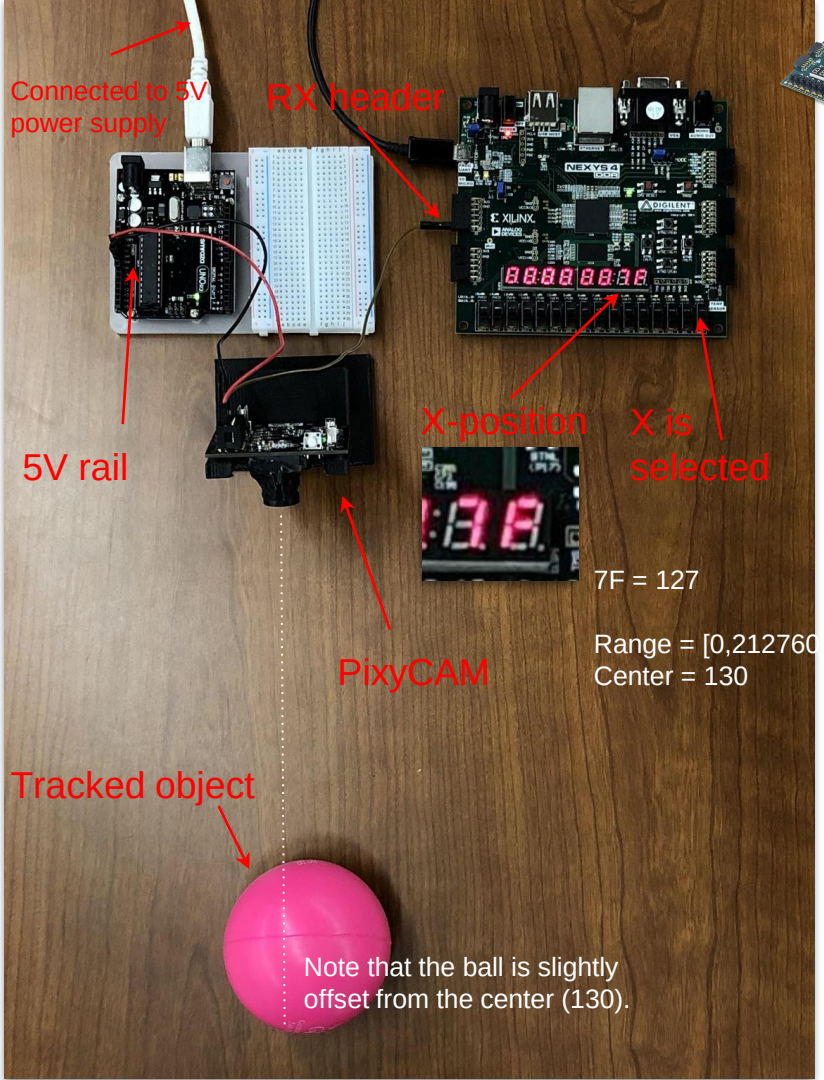
1. Tracking objects
2. Projectile motion estimations
3. Obstacle avoidance

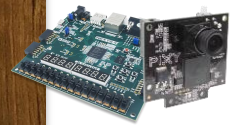
Application example: an autonomous rover that retrieves an object and returns it to a designated drop-off location. (2800 Project of team member).

EXPERIMENTAL SETUP

The PixyCAM is calibrated with the accompanying software PixyMON, where a signature can be declared through the viewport of the camera, the sensor is then further calibrated (white-balance, exposure, etc.)

Because the PixyCAM requires a minimum of a 5V power supply and the FPGA only has 3.3V rails, an Arduino is used to serve as the power supply. The PixyCAM TX pin was connected to the JD1 header of the FPGA, which serves as the RX on the UART.





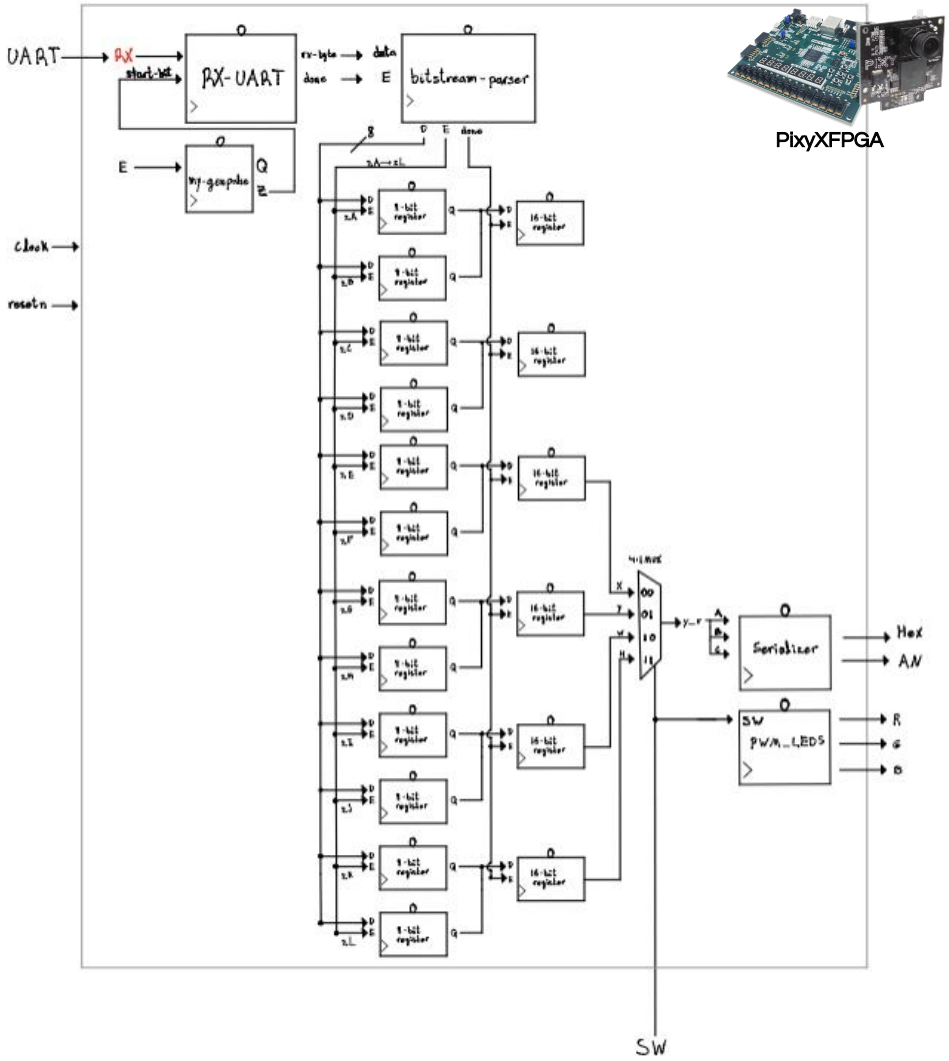
TOP LEVEL DIAGRAM

The VHDL program parses the RX signal's bits via the component RX_UART. The data is then re-compiled into bytes by the bitstream-parser. The genpulse acts as a bit counter (@ 19200 baud).

The bitstream-parser enables 8-bit registers (zA => zL) which hold the compiled bytes.

Once the registers have received their data, a 'done' signal is sent from the bitstream-parser to enable the array 16-bit registers, which combine the bytes held in the 8-bit registers into 16-bit words.

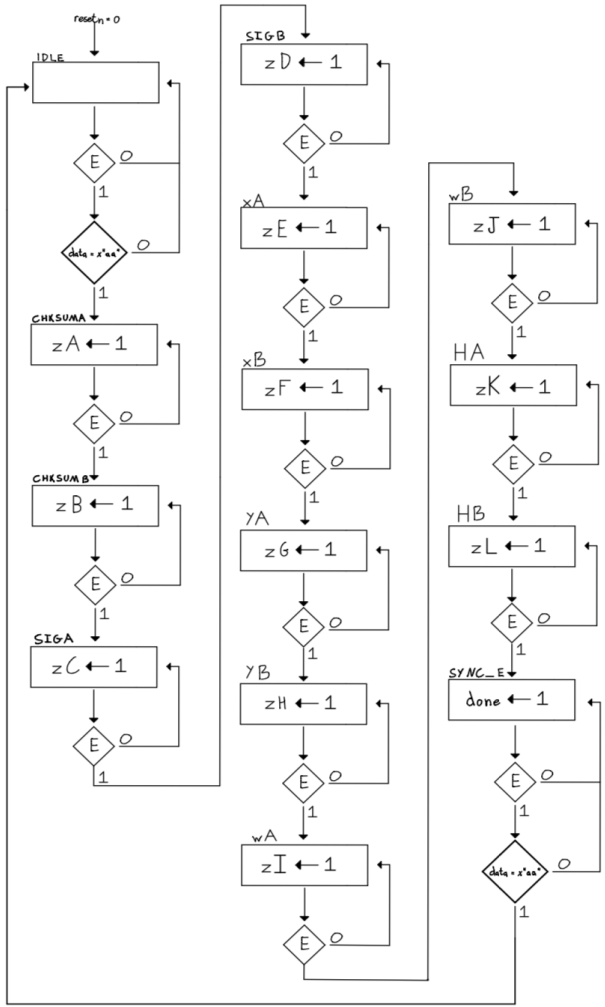
Switches determine which value is displayed on the 7-seg display; the onboard RGB LEDs displays a color to indicate which value is being shown (yellow->x, blue->y, magenta->w, green->h)



PixyXFPGA

Finite State Machine Diagram

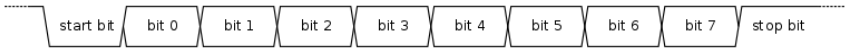
The FSM is to activate enable signals for registers to store the data from the UART RX to their corresponding signals as they are transferred in.



RX_UART COMPONENT

Data Stream:
The data coming from the PixyCAM uses the UART protocol. In order to use the PixyCAM object tracking the signal needs to be translated into a usable form.

Baud Rate and Sample Rate
The baud rate is set on the PixyCAM (in this case, it is 19200 bits per second). To oversample the signal a frequency of 16 times the baud rate (307 KHz) must be used in a counter.



Example of a UART data stream

Sampling:
Interpreting the UART signal can be done by oversampling the signal in a FSM. To do this a start signal needs to be generated at 16 times the baud rate. A counter is incremented, starting from zero, every time the start signal is high until it reaches 7 (the center of the start bit). The counter is cleared and the state is moved to data collection mode. Now that center has been detected the start signal is watched 16 times to reach the center of the first data bit. The value is stored and the process repeats for the number of data bits in the given signal (in this case 8 bits). Once data collection is complete the stored data is assigned to an output and waits 16 ticks so that it is in the stop bit to wait for new data.

TESTBENCH

PixyXFPGA