

PI Cam

Raspberry Pi controlled RC helicopter

Team Bucuresti_2

1. Title: PI Cam

2. Team

- **Liviu-Alexandru Rădoi**: hardware design and implementation
- **Antonel-George Dobre**: Raspberry PI software design and implementation
- **Andrei-Alexandru Musat**: Communication protocol and component integration
- **Ioan Deaconu**: Mobile application developer and hardware interfacing

3. Brief Introduction

This year's competition is a challenging, and also thrilling experience. The combination of image detection software together with the automatic control of a toy helicopter represents an interesting project that we are more than willing to tackle.

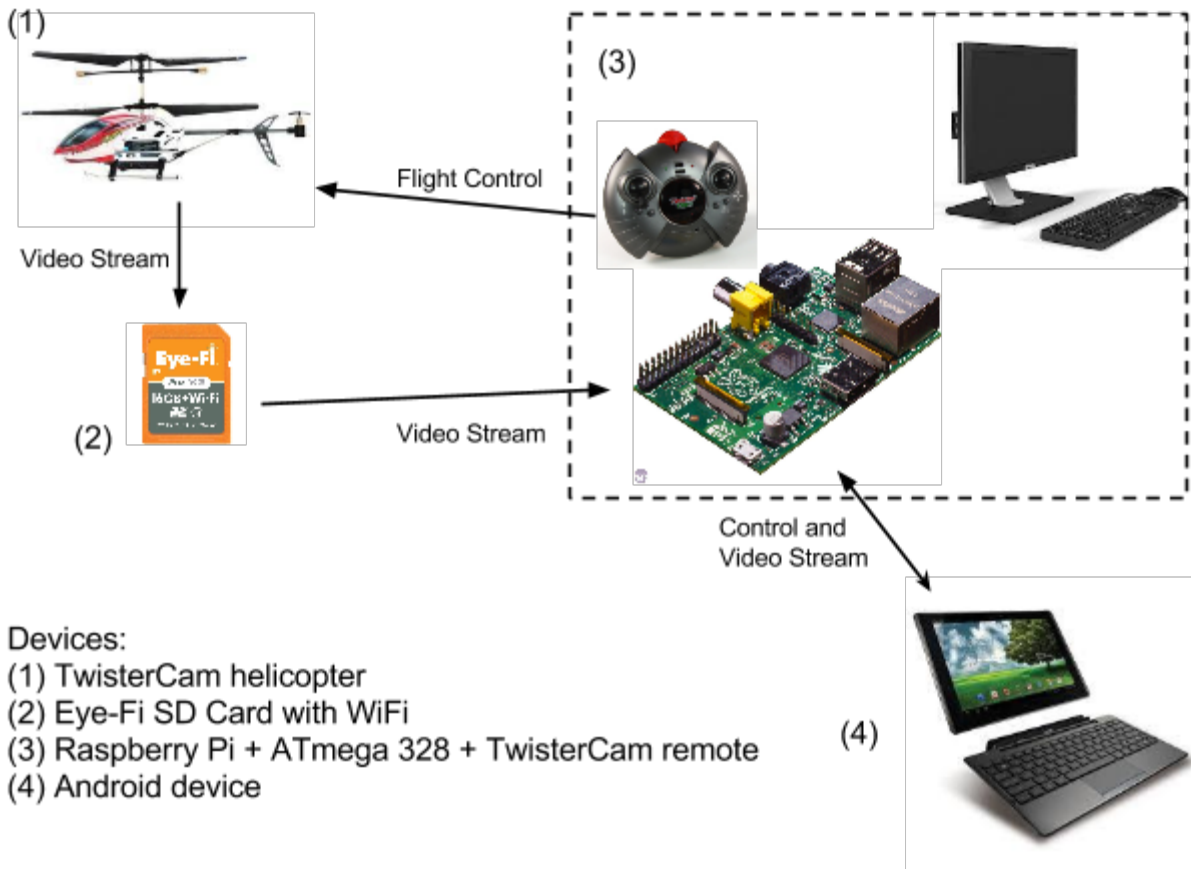
In order to keep up to the task at hand, our team has decided to take a basic route and try to understand how the basic components that were given to us work (the TwisterCam helicopter and its remote control, and the Raspberry PI). Gradually, we understood the magnitude of the project and the difficulties we would have to overcome in order to realize our project.

First of all we have to control the helicopter without the need of the remote control. *Second*, and probably our biggest challenge, is to stream the contents of the microSD card that contained the camera footage towards our mobile control app, or any remote client that wished to connect to the airplane. *Third*, we had to detect the shapes from the images of that footage in order to see our track. *Fourth*, all of this has to happen autonomously.

But we manage to come up with ideas and ways to work around our limitations and try to resolve the tasks in the shortest time possible. In the following chapters we will describe our take on each of the discovered challenges and what the results are. Our progress, based on each day of activity, can be seen on the bottom of this paper.

4. System Description

Our approach towards solving this year's tasks revolves around the integration of multiple platforms and components in order to properly operate the given RC helicopter.



(1) TwisterCam helicopter

The TwisterCam helicopter, as supplied by the organisers, includes a video camera used in our system for video streaming and marker detection and recognition. Moreover, to provide simple autonomous control of the helicopter, we have adapted the remote control to work with the Raspberry Pi.

(2) Eye-Fi SD Card with WiFi

The [Eye-Fi SD card with WiFi](#) is a memory card bundled with a WiFi transceiver, created for easy download of photos and videos for consumer cameras. However, using it in our system proves to be challenging, due to problems with supplying power and implementing interrupts on the camera to enable a start-stop procedure for low-latency video streaming.

(3) Raspberry Pi + ATmega328 + TwisterCam remote

The “brain” of our entire system is the Raspberry Pi Single Board Computer. The Pi is responsible for keeping our system up and running, receiving the live video stream from

the helicopter's camera and sending commands to the helicopter after processing the image feed. In order to perform its tasks, the Raspberry Pi runs a series of software programmes with various functionalities, written in different programming languages:

- a shape detection algorithm (written C/C++ using OpenCV libraries) using edge algorithm, which can find various geometrical shapes in an AVI format video;
- a C-based server which accepts multiple incoming connections and communicates with its clients via a custom protocol. The messages that are received by the server contain data necessary to control the ATmega328 microprocessor on the attached Gertboard Extension;
- by combining these two functionalities into a state machine algorithm, the Raspberry Pi will also be able to control the helicopter autonomously, without the need of a human operating the remote control;

Added to the Raspberry Pi in our system is the Atmega 328 controller on the Gertboard. This could be considered our system's motion center. It receives information from the Pi, processes it and generates PWM signals. These signals are used to simulate the output of the remote control's varistors, thus no longer needing mechanical interaction to control the helicopter.

(4) Android device

The Android device provides rich user interface. The Android application we developed serves two main functions: (a) displays video stream captured from the camera and (b) can serve as remote control if the user wants it, duplicating the role of the initial remote control, that we adapted to work inside of our system.

5. Current Status

Both hardware and software development is undergoing at full throttle:

a. Hardware

- integration of the Raspberry Pi with the ATmeta328 and with the remote control are done, and commands from the Raspberry Pi can be sent to the helicopter
- experiments to give power to the Eye.Fi card are undergoing

b. Software

- processing AVI files to detect and recognize markers on Raspberry Pi is done in a C application that uses OpenCV
- a server running on the Raspberry Pi is listening for commands coming from the Android device; we have initially written this server in Python, but we ported it to C for increased performance
- ongoing work for a server on the Raspberry Pi to capture and forward the video stream

6. Challenges and platform limitations

This section details the challenges we have faced and our proposed solutions:

- The helicopter is very small and can not carry a lot of extra weight. Therefore not much hardware was added on the helicopter, all processing hardware is fixed and we implement wireless communication for all features;
- The remote control for helicopter is using an unknown protocol sent over a 2.4GHz wireless network. The protocol seemed to be similar to SPI. The solution used for this problem was to replace the input control from a varistor, responsible for changing the helicopter's position, with a pwm channel.
- In order to control the remote from Raspberry, we needed 3 PWM channels. The Raspberry board has only one hardware PWM channel and the rest of the channels are simulated in software. One of the biggest disadvantages of using software PWM is that it needs a separate thread which consumes a significant amount of resources. In order to pass over this limitation, we used a Gertboard containing an Atmel ATmega328 microcontroller and three of it's hardware PWM channels.
- The camera integrated in the helicopter is saving the data file on a micro-sd card placed underneath it, while, the wireless streaming of the video output needed to be implemented by us; after a lot of re
- Shape detection is difficult on a Raspberry Pi because of the limitation in computing power
- python server too slow, ported to C
- parts falling (a iesit axul de la greutatile de la elice) from TwisterCam helicopter
- being a wireless card, Eye-Fi consumes a lot of power, and because of that, the circuit goes into protection and shuts down the entire board

7. Day by day journal:

Monday:

- brainstorming

Tuesday:

-

Wednesday:

- ported the Python server on Raspberry Pi to C
- integrating Eye.Fi into the system (power supply, interrupts on camera)

Thursday:

- Helicopter automated flight testing
- Finalizing the Machine Learning component of the software system

Friday:

- Minor bug fixes and routine testing