CHAPTER 2

LITERATURE REVIEW

2.1 Reviews from Other Projects

There are numerous projects that are associated with this project. To make things easier, this section will focus on few projects that are heavily associated with this project. One of them is Stanford Aerospace Robotic Laboratory [1]; The ARL has been working for many years on technologies based on aerial robotic control. These technologies include:

- "Object-Based Task-level Control" – allowing the user of an automated system to command a task, rather than be consumed with low-level issues of the task
- GPS for real time control
- Vision for real time control
- Path planning in structured and semi-structured environment
- Real time "on-line" system identification.

Competitions had been organized under the same concept. One of such competitions is the international aerial robotic competition that started since 1995. The objective of the competition is to demonstrate a fully autonomous air vehicle which can:

- Take off from a 15' square given location
- Over fly a 120' by 60' grass field without crossing the boundaries
- Determine the location of five 55-gallon plastic barrels
- Identify the type of barrel as either biohazard, radioactive, or picric acid
- Find and retrieve a ferromagnetic disk, located at one of the barrels.

- Return and land safely at the take off location.

Other university had also done the research based on the same concept. Below show the research and development done by University of Carnegie Mellon in the recent years [3]:

September 1991

Initial attitude control experiments

- Attitude control test bed developed.
- Used to test and tune attitude control system.
- Electrical model helicopter mounted on a swiveling arm platform.
- Optical encoder mounted with frictionless bearing measures ground-truth angles in real time.
- Configurable for roll, pitch, and yaw.

February 1992

Free flight and vision-based state estimator

- Six-degree-of-freedom test bed developed for evaluating various position estimation and control systems.
- Electrical model helicopter attached to poles by graphite rods for safety and helicopter ground-truth position estimation.
- Lightweight composite material and custom designed frictionless air bearings allow unobtrusive helicopter free flight in a cone shaped area.
- Mechanical stops prevent the helicopter from crashing or flying away.

September 1994

First autonomous platform

- Indoor test bed developed as an intermediate step towards autonomous operation.
- Used for testing the vision system, control system, sensor platform, power system, RF interference, and overall system integrity.
- Allows relatively large (1.5 meter) longitudinal travel.

- Severely limits helicopter travel laterally and vertically.
- Helicopter is tethered with ropes which are fastened to the ground and two poles positioned on either side of the platform.
- Steel rod with hooks on either end connects the ropes to the helicopter.
- Steel rod is secured to the helicopter's center of gravity to eliminate any torques from restraining forces which could cause dangerous rotations.
-

October 1995

Autonomous Helicopter #1

- Visual odometer (4 cm accuracy, 60Hz update), tracks image patches and templates with helicopter motion.
- Initial computer control trials performed at relatively high (~15m) altitudes to allow safety pilot time to override computer.
- Latitudinal and longitudinal controls were tested first by mixing human control for height and heading with the computer commands.
- Heading and height control were enabled as the computer control proved effective in stabilizing the helicopter
- GPS used for ground-truth measurements.

August 1996

Autonomous Helicopter #2

- Control system for autonomous takeoff, landing and smooth trajectory following.
- System tested in harsh conditions (40-45 mph wind gusts).
- State estimator fusing data from a dual-frequency carrier-phase GPS receiver, 3-axis angular rate and inertial sensors, and field-rate vision-based odometer.

- Custom-designed vision system capable of field-rate position sensing, multiple object tracking, color discrimination, and aerial intensity map building.
- Custom-designed camera stabilization system.
- 3D laser line scanner.
- Power system for up to 33 minutes autonomous operation.

July 1998

Autonomous Helicopter #3

- On-board laser mapping system
- Deployed in Haughton Crater, Devon Island NWT Canada

As mentioned before, this project's objective is to enable the data from the camera to be sent back to the PC based station via RF signal. While the above mentioned projects (Stanford Aerospace Robotic Laboratory and Carnegie Mellon aerial Robotic) have many features that this project currently do not possess, the concept of aerial robotic and the usage of camera system is the same, and this make these projects worth to be review. To have a better understanding of this project and also to achieve the objectives of this project; one of the projects by Wolf Paulus, an experienced software developer and innovator were review and studied. Wolf Paulus's project of low resolution color camera driver for 8031-SDK are in fact one of the main inspiration and source of information for this project.

Wolf Paulus's main objective is to demonstrate how to interface a low resolution camera which is equipped with a serial port, with an 8032SDK microcontroller [4]. A CMOS low resolution color camera's RS232 serial interface is connected to the SDK internal serial board. This connection is to allow SDK to download raw image data from the camera into its SRAM and also to control the camera. The following section will elaborate in details regarding Wolf Paulus's project.

2.2 Review from Other Project: Low resolution Camera Driver for 8031-SDK

In this research paper, a CMOS-based low-resolution (160 × 120 pixel) color camera's RS-232 serial interface is connected to the SDK's internal serial port. This connection is used to allow the SDK to control the camera and to download raw image data from the camera into the SDK's SRAM. The SDK's external serial port is connected to a PC/Terminal, providing a basic user interface, which allows for user controlled processing.

The low-cost digital camera does not provide any image processing and therefore, after a photo has been taken, about 20 Kbytes uncompressed, raw image sensor data, is downloaded into the SDK's SRAM memory. Since the camera's serial interface only support a single transfer speed of 57,600 baud, optimized assembler routines had to be implemented to allow the SDK to receive data and immediately transfer it into external memory at that speed. About 20 Kbytes are needed to store the raw image data for one image.

The readout happens in a horizontally shuffled Bayer colorization pattern and requires quite a bit of processing before a 24-bit per pixel MS-Windows compatible bitmap can be created.

The full QSIF standardized bitmap (160 x 120 pixels with 24-bit color information), about 56 Kbytes, doesn't fit into the available external memory. Therefore, the image data is sent out through the external serial port while it is processed. Before the first pixel can be send however, a MS-Windows compatible BMP header is sent.

For the lack of availability of a better protocol, the receiving terminal has to log the receiving data, to eventually being able to display the bitmap. That however requires the SDK to send the data UUEncoded (standard for data interchange between systems with possibly different code sets, and to represent binary data as a text file).

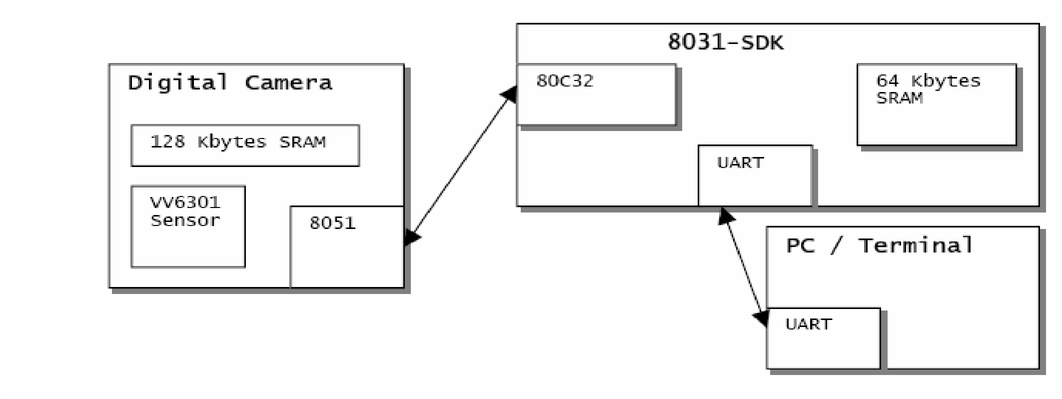

Figure 2.1: Hardware Block diagram for Wolf Paulos's Project [1]

2.2.1 Hardware Design

The Camera hardware consists of the following major components:

- STMicroelectronics VV6301 CMS Sensor
- TEMIC 51 X3702/B (Intel MSC51 compatible) Micro Controller
- Winbond 128 Kbytes CMOS SRAM
- 2 TI 97E1L1M High Speed Counter

2.2.2 Software Design

To support re-use and better maintainability of the software, the following modularization was chosen:

- **Main.c** contains the loop that waits for and handles user inputs.
- **CamDriver.c** contains the camera specific functions, allowing control of the camera.
- **ImgProc.c** contains the implementation of the Bayer-decoding algorithm.
- **UARTex.c** contains replacements for STDIO lib function to allow I/O through the SDK's external UART

• **Serialin57.asm, Serialout57.asm**, high-speed serial driver for the SDK's internal serial port.

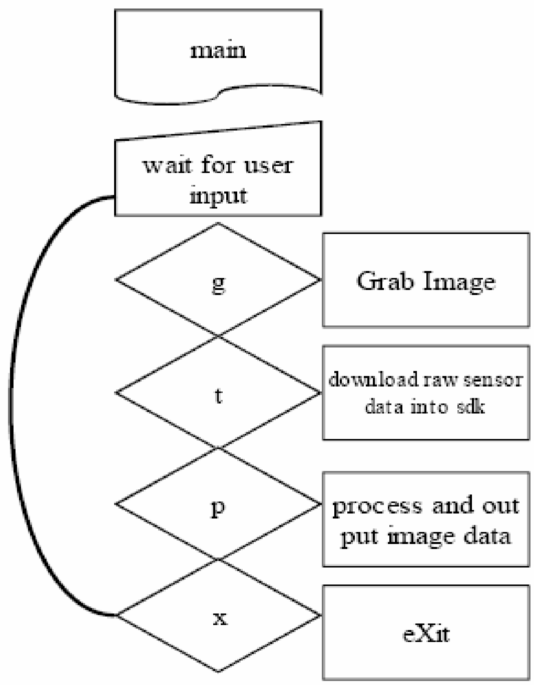

Figure 2.2: Software Flowcharts [1]

2.2.3 Serial Communication

Serial communication between the SDK and the PC through the SDK's external serial port at 9,600 to 38,400 baud and since the camera can only support a single connection speed at 56,000 baud. Serial port driver have to be develop. Calculation are needed to sample each incoming bit once

Timer Calculation for 57,600 baud:
11.0592MHz / (12*57,600Hz) = 16 and 256-16= 240 = **F0h**

2.2.4 Camera Driver

Protocol

All communications with the camera has to be initiated by the connected 80C32 controller,

A command from the controller has the following format:

STXCommand Data 1Data 2…Data NETX

The response has the following format:

STXResponseData 1Data 2…Data NETX

Example: Grab Image:

Cmd: 0x02 G 0x03, Response: 0x02 g 0x03

The complete image is returned as a stream of data bytes, including black lines, visible lines and status bytes.

2.2.5 Image Processing

Wolf Paulus's project covered until the image processing part while for this project, its objective is to send the image data back to the PC based station; it is still worth mentioning that Bayer Pattern concept is used in the image processing part. The Bayer Pattern is based on the premise that the human eye derives most of the luminance data from the green content of a scene; and it is the resolution of this luminance data that is perceived as a the "resolution" of an image. Therefore, by ensuring that more of the pixels are "green", a higher resolution image can be created - compared with an alternating R-G-B color filter array with equal numbers of Red, Green and Blue pixels. Information on this sub-chapter is quoted from reference no [1] in Reference section.

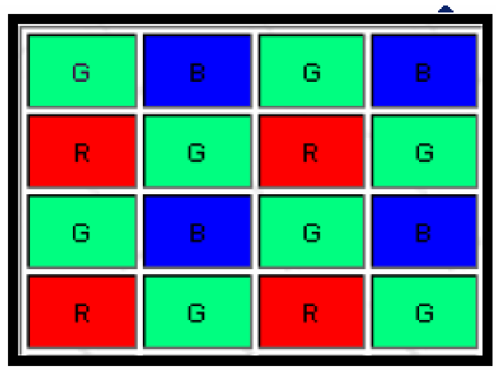

Figure 2.3: Typical Bayer Patterns [1]

2.3 Review of Other Product: Webcam Application Expansion Board

Other from projects, there are one worthy product in the market that will be mentioned here as this product has a lot of similarity with this project [5]. 8051EvaluationBoard is a Webcam application expansion board embedded with i2Chip W3100A, a hardwired TCP/IP chip ("W3100A"), integrated with CMOS-type camera to transfer video data over the Internet without any PC. Its main components are the following: Camera Sensor, M-JPEG CODEC, Memory and an interface with 8051 EVB. 8051 Evaluation Board is comprised of 8051 MCU, Memory, W3100A (TCP/IP), RTL8201(Ethernet PHY) and an interface with Web Camera Module. Figure 2.4 below shows the product, webcam application expansion board:

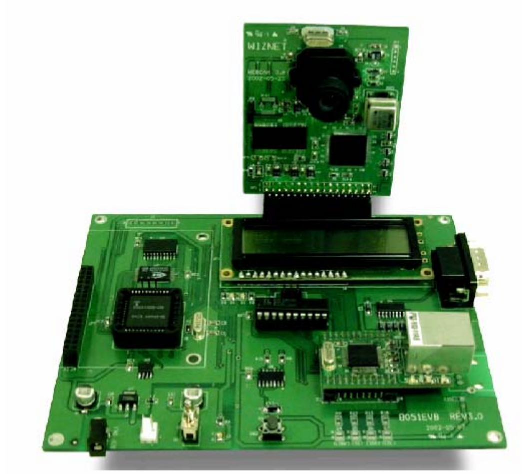

Figure 2.4: Webcam Application Expansion Board [5]

2.3.1 Operation of the Webcam Application Expansion Board

8051 MCU will initialize the internal registries of CMOS Camera Chip (OV7620) and then transmits the setting information on jpeg frame to M-JPEG Chip (LC82210). CMOS Camera Chip will transfers the video information received through Lens and Sensor to M-JPEC Chip in YUV Format. At this time, CMOS Camera Chip and MJPEG Chip need to be synchronized. Synchronization corresponds to the ZV Port Timing and proceeds as follows. Following the falling edge of VSYNC (Vertical sync pulse) signal, as HREF (Horizontal valid data output window) transforms from rising edge to falling edge (while at high), YUV signal will be considered as Valid Data, and the YUV signal is transmitted according to PCLK where 8bit Y signal and 8bit UV signal are combined to 16bit each.

Jpeg frame is created upon request from 8051 MCU to M-JPEG Chip, and then M-JPEG Chip saves the jpeg frame in DRAM and sends the Interrupt to 8051 MCU regarding the data. When the Interrupt is received, 8051 MCU receives the jpeg frame from MJPEG Chip and saves the data in SRAM of 8051 Evaluation Board. At this time, 2 frames are saved in SRAM based on Double Buffering method.

When a request for jpeg frame is received, 8051 MCU reads the jpeg frame from SRAM and transfers the data to i2Chip W3100A, and then i2Chip W3100A transmits the data to the remote PC.

To have a better understanding of the board, Figure 2.5 below will show the block diagram of the Webcam application expansion board.

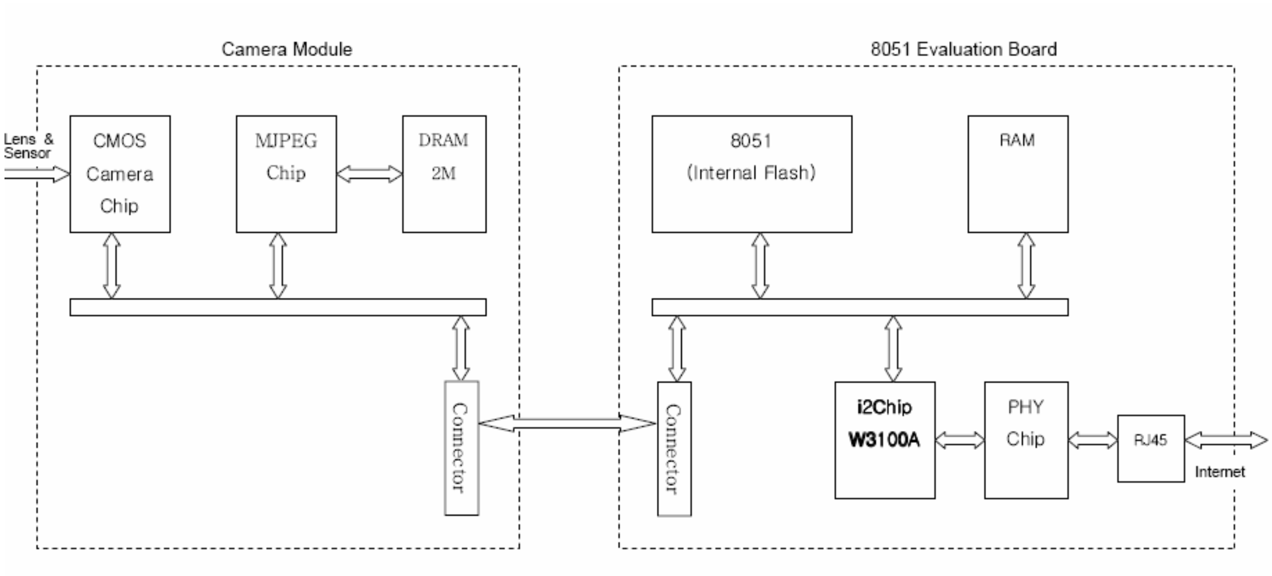

Figure 2.5: Block Diagram of the Webcam Application Expansion Board [5]

2.4 Reviews from Other Project: Wireless Roaming Web Cam

One particular project that use wireless communication to transfer data (via RF communication) is a project named Wireless Roaming Webcam [6]. This wireless roaming webcam project proves to have a lot of similarity in term of concept and also the implementation process of this very project. To sum it out, three main criteria that have great similarity if not the same concept were review in this section.

In this Wireless Web Cam project, it involves moving a robot, which transmits 128 x 128 pixel images to a PC, via a Graphical user Interface (GUI). Communication between the robot and the PC in this project is achieved using two RF links, one dedicated to the transmission of movement commands to the robot and the other dedicated to the image transmission. While this project involves a lot of mechanical parts, it has three main criteria as mentioned before, that need to be addressed. The three criteria are the camera subsystem and the wireless data transfer.

2.4.1 Camera Subsystem

This camera subsystem used Mitsubishi Artificial Retina (M64282FP), this camera is able to produce 128 x 128 pixel grayscale picture and it has built in image processing and analog output tuning functions.

The goal of the camera subsystem was to be able to read images from the camera and send them to the RF transceiver subsystem to be relayed to the PC. Images are read from the camera and sent on at a rate of approximately 0.3 – 0.5 fps (frames per second). The ADC1061 Analog to Digital converter (ADC) was used because it is a flash type so its conversion time is quite fast. Successive approximation type ADC's employ less complicated circuitry at the expense of conversion time. It is clear in this case that a fast ADC is required in this system. The Figure 2.6 below show the camera interface system

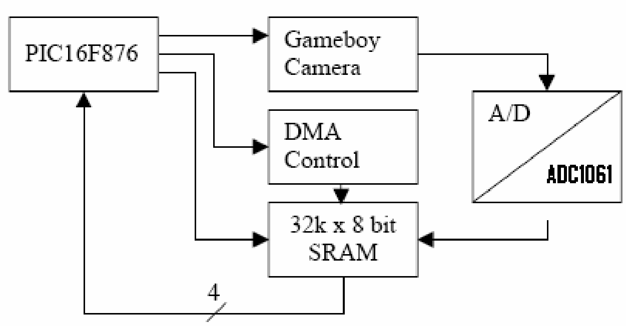

Figure 2.6: Camera Interface System [6]

2.4.2 Wireless Data Transfer

Two RF links are being used for this project. The RF modules being used are the Linx Technologies RM-418/433 MHz transmitters and receivers. The 418MHz link does not interfere with the correct operation of the 433MHz link and vice versa. Since the receiver is always receiving a signal regardless of the state of the transmitter, a packet header is added to the bit stream to filter out valid from invalid data.

There are two complete transceiver subsystems used in the project. One is attached to the serial port on the host PC. This functions to receive data on the 418 MHz carrier from the camera subsystem and send data on the 433 MHz carrier to the motion control subsystem. The other is attached to the robot and is used to transmit image data from the camera subsystem (418 MHz) and to receive data (433 MHz).
The data can be transmitted at 9600 bits per second (bps). This results in a frame rate of approximately 0.3 to 0.5 fps

In the paper, a suggestion is given to have some higher bandwidth RF link implemented. If this were done, then a higher frame rate could be achieved with the video. If, for example, a 57,600 bps transmitter/receiver pair was used, then the frame rate up to 3 fps can be achieved. The maximum range that the RF can be archive according to this paper is 15m.

2.4.1 User Interface

The function of the software is to display the image sent by the camera through RF, and to transmit signal to control the robot as well as camera. Consequently, the two main tasks of the software implementation involved the image processing unit and the interface unit. Figure 2.7 on the next page shows the software application of the system. All the information on this subchapter is found in reference no [6] in Reference section.

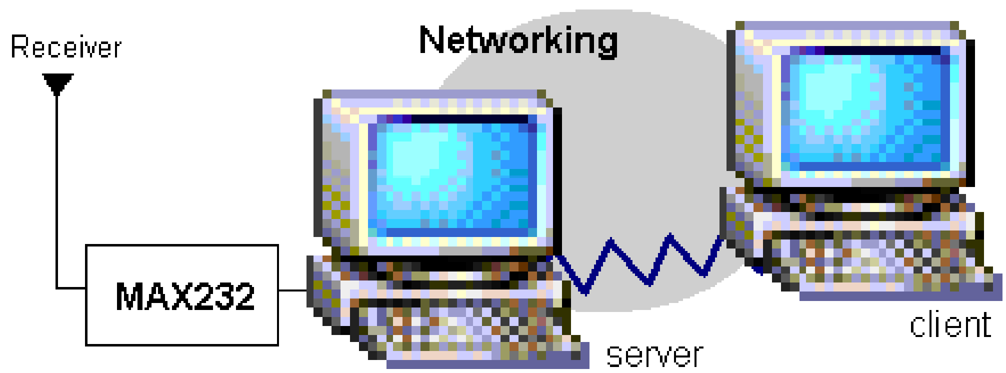

Figure 2.7: Software Applications [6]