

EE 318 Electronic Design Lab-1 project report

Surveillance Robot

Group D3

Shreyans Gandhi (06d07005)

Mohit Dandekar (06d07006)

Praveen Tamhankar (06d07007)



Guide : Prof. Vivek Agarwal

Department of Electrical Engineering

IIT Bombay

April 2009

Abstract

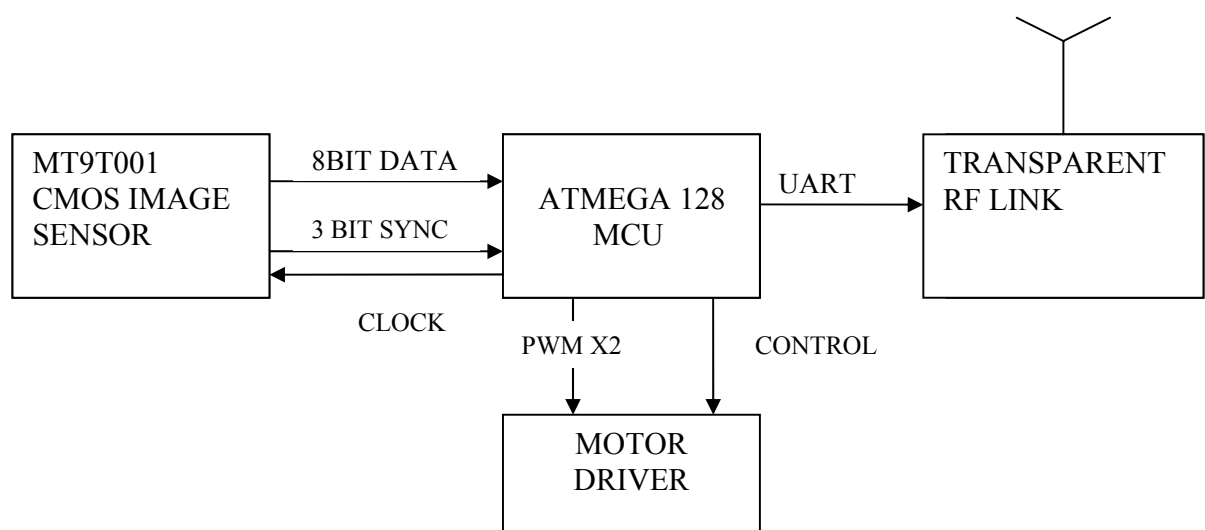
Our project is to design and build a manually controlled surveillance robot. The main purpose of the robot is to be able to roam around in a given environment while transmitting back real time data (video) to the ground station. This real time data can then be used by the controller (human) to move the robot around. The robot must be compact and self contained with wireless transmission of data.

Introduction

The main purpose of the robot we are making is to provide visual information of hard to access places, for example a building under a hostage situation. Hence the main feature of our robot is an onboard video camera. Also the robot must be compact and self contained in the sense it must have an onboard battery pack and wireless interface to the human controller.

System Description

Robot Onboard System



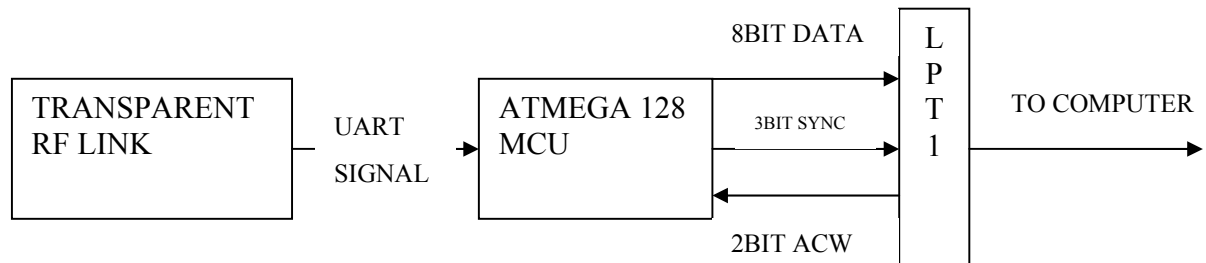


FIG 1

Robot Onboard System

The robot onboard system consists of the MT9T001 CMOS image sensor , ATMEGA128 microcontroller to interface to the camera and process the captured image, a transparent RF link, and motor driver and actuator for control of the robot.

CMOS IMAGE SENSOR:

- The image sensor chosen is the micron MT9T001 CMOS image sensor that operates at 1 Mhz clock input.
- The clock to the camera is generated by the onboard microcontroller utilizing the waveform generation mode.

The synchronization logic to interface the camera is as follows :

- There are three sync signals, namely FSYNC (frame sync) , LSYNC (line sync) , PXCLK (pixel clock)

- The frame sync signal is to signify the start of a new frame, in default mode this signal has a rising edge at the start of the new frame, and it remains high as long as the pixels in the frame are being shifted out.
- In between two frames the fsync signal goes low for a short duration and remains low till next frame starts signifying frame end.
- The line sync signal behaves in a similar fashion as the frame sync.
- The pixel clock signal is used to shift out pixels one by one. The pixel data is latched onto the data port of the camera every rising edge of the clock.

The pixel shift out procedure follows the following logic: Pixel are read out in a horizontal manner, i.e. pixels in one line are read out first followed by a transition in LSYNC signifying and then the new line is read out left to right.

The data rate (pixel shift out rate), the window size are programmable entities. There is a serial (i2c) interface to the camera that enables programming of the camera, however in our case the serial link is malfunctioning so to achieve desired window size of 48 x 48 pixels we used the following technique:

- The camera by default shifts out 2048 x 1536 pixel image window out of which we are interested in only 48 x 48 pixels, hence on the controller side we ignore the rest of the lines / pixels and only pick up the desired pixels.
- The window size of 48 x 48 is arising from the limitation that we do not have much of internal SRAM in the microcontroller to store the captured image.
- The solution to that was to interface serial RAM or SD card that can provide adequate storage space, however we were not able to do so due to lack of time.

MT9T001 CMOS IMAGE SENSOR

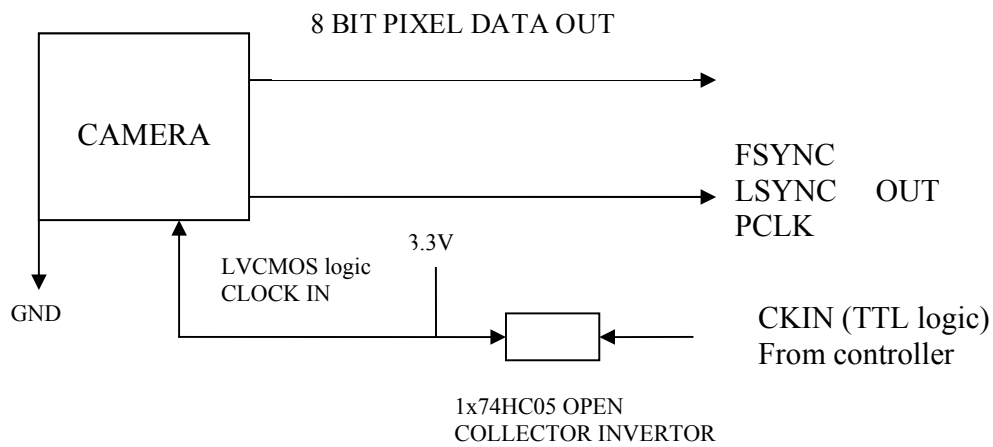


Fig 2

MCU: ATMEGA128

The onboard microcontroller is the AVR series ATMEGA128, for maximum throughput we are running the controller at 16 MHz.

There are three main functions running on the controller: `acquire_data()` , `compress()`, `transmit_data()` . the software block diagram is in figure 4.

`Acquire_data()` : fig 3

CAMERA INTERFACE ALGORITHM

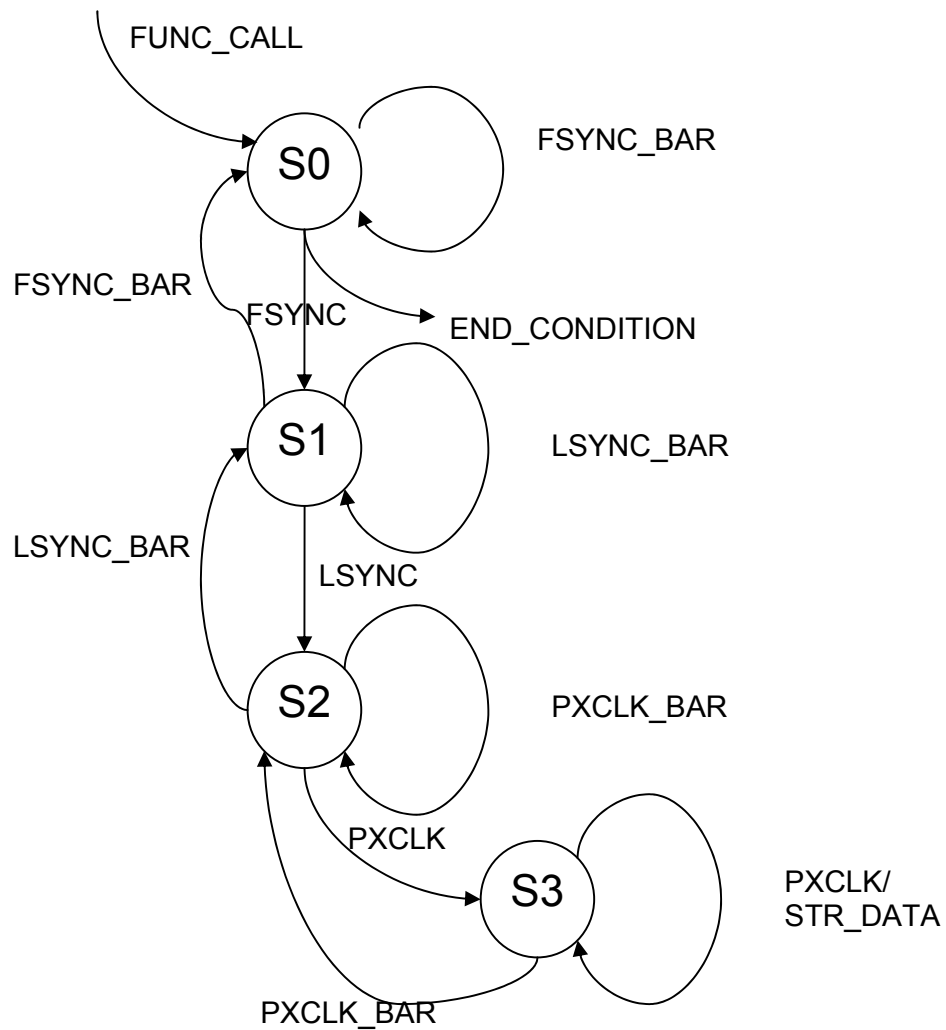


FIG 3

The transmit data function simply shifts out the compressed data through UART. The UART txd pin is connected to a transparent RF link that repeats the signal and is connected to the rxd pin of the base station microcontroller.

Due to lack of time we could not import the MATLAB version of the compression routine to the microcontroller.

Software description

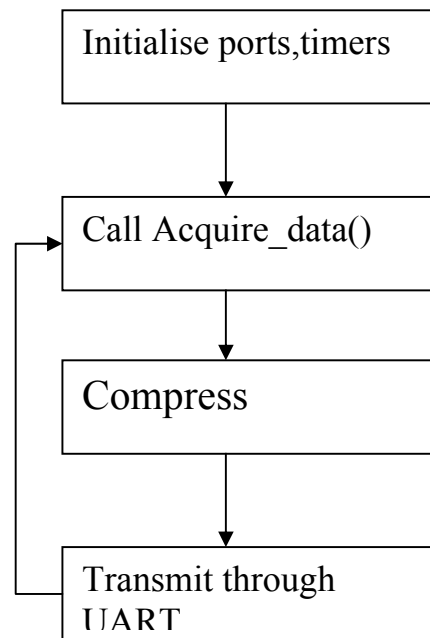
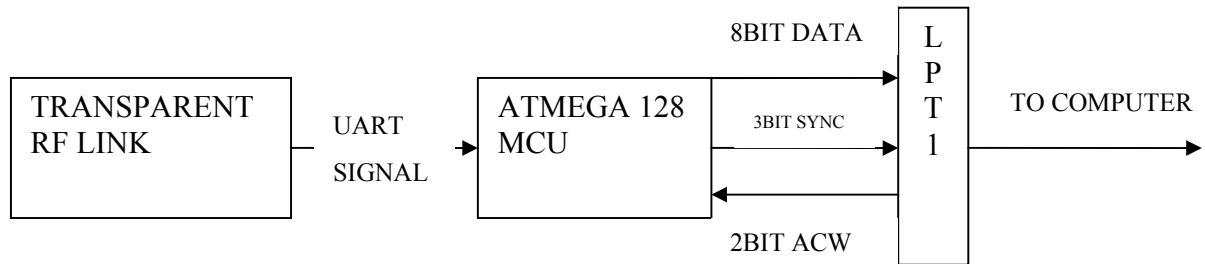


Figure 4

Base Station:

The base station does the reverse of the operation done on the robot mcu. Although we have not implemented the compression routine but in case we had the base station decompresses the image data and uploads it onto the computer through LPT1 parallel port.

The image plotting on the computer is done using OpenGL, and Parapin library is used for interfacing the parallel port.



Limitation of our work

- The RF link in place is very slow with a baud rate of 2400 bps however the required baud-rate for 1 fps 19200 bps
- 48x48 pixels is a very small image size, not enough for any meaningful display. For supporting larger image size external RAM was required.
- Compression algorithm was not implemented on microcontroller (though it was tested)

- Camera interfacing could not be done correctly, the result was waste of time as we were only using 48x48 pixels out of 2048x1536 pixels shifted out in default mode.