EE318 Electronic Design Lab Project Report, EE Dept, IIT Bombay, March 2007

Interface between a Camera and a MOTE

Group B-8

Mukul Shakya (04007008)
Asad Ali (04007009)
Ankit Jain (04007014)

Supervisor: Prof. S. N. Merchant

Course instructors
Prof H, Narayan
Prof D. K. Sharma
Prof M. Bellur

Abstract

The report discusses the design of a circuit that acts as an interface between a camera and a MOTE to control the transfer of data between the two devices. The need of the circuit arises from the fact that the mote works at a speed much less that that of the camera. Also the MOTE only accepts the packetized data whereas the camera sends 8 bit parallel data output. Hence the interface stores the data coming form the camera (at 27 MHz) in the FIFO memory, then packetizes it and sends it to the MOTE (which works at around 38.6 Kbps). The MOTE in turn transmits data to the receiver.

Problem statement

To design an electronic circuit that can receive 8 bit parallel output data coming from a camera, store it in a memory, packetize the data and send it to the MOTE at around 38 Mbps.

- We have a camera which sends 8 bit digital output. To achieve the same, we have used an analog camera and have converted the output in digital format using a TVP5150A chip and an associated circuit. The output data comes at 27MHz.

- Also we have a MOTE which transmits and receives data at 38.6 Kbps. It sends data to a USB supported device. It uses WSN protocol and uses

RS232 for communication. It receives and transmits data in packetized format, the start byte of which is identified by the presence of 7EH byte. The maximum storage capacity of the mote is 20K and it receives and transmits data in a continuous fashion. It communicates with the microcontroller via the I2C protocol.

Introduction

Of all the surveillance operations that are being carried out nowadays not all require a constant monitoring in the form of real time video data. In some cases, still image data at a fixed interval of time will be sufficient. In such cases, using video cameras will be expensive as well as more power consuming. A better solution will be to capture image data at fixed interval of times and transmit it to a remote receiver via a MOTE. Unfortunately, there is a stark difference in the sending data rate of a normal camera and the corresponding receiving data rate of the mote. Also the MOTE receives data in a fixed format that may not be provided by all the cameras available. So the MOTE cannot be directly connected to the camera.

Hence we made a circuit that acts as an interface between the two and provides a cheap solution to an otherwise expensive job.

The circuit needs to have the following specifications:

- It must have a memory buffer of at least 300k or must have memory that can be cascaded to provide this storage capacity.
- It should be able to read at 27 MHz and write at 38.6 Kbps.
- It should be able to send the data in a format that the mote can read.
- It should be cost efficient and consume as low power as possible.

Design Approach

The circuit requirements as analyzed in the introduction prompted us to think of laying the 1st preference on designing the circuit to transmit data from the FIFO buffer to the MOTE. After successfully transmitting the data to the MOTE, we directed our attention to the storage of data from the camera to the mote. The steps undertaken were as follows:

- To check the sending and receiving of data from a computer terminal to the microcontroller via the UART.

- To store junk data in the FIFO memory and check the transmission of it on the computer.

- Now to send the junk data to the MOTE after packetizing it in the microcontroller.

- Then storing the data from the analog camera in the buffer and transmitting it to the MOTE.

The **hardware** design was planned as follows:

1. To obtain a circuit for converting the analog output of the camera into digital format. TVP chip was used for this purpose.
2. To make the standard circuit for the microcontroller and using an adaptor board, connect the flat pin buffer to the circuit.
3. Connect the µC to the mote via the UART.
4. Connect the camera part to the microcontroller.

For the **software** part, the steps were undertaken as previously stated. The µC was coded to read the data from the FIFO and packetize it. The maximum length of the packet is 255 bytes. The exact details of this scheme are given in the references.
The remaining task was to read data from the memory buffer which was done using the usual algorithms.

Design of Circuit

The options we had in front of us initially were:

- Using the ATMEGA 128L microprocessor since it had a very good speed that could allow us to perform additional tasks like sending of only the error function instead of the entire image. This would have been quiet energy efficient.

- Using a simple 89c51 microcontroller instead because of its simplicity, cost and easy availability.

- Using an analog camera and then converting its data into digital format using TVP chip or directly using the digital camera.

- What kind of memory buffer is to be used. We had considered Averlogic 422b, Renesas, Hitachi and TI chips.

- Whether to consider including the comparing of images feature in the interface which could have been useful in saving power consumption in the areas where there is not much activity expected e.g. near a safe vault or at traffic signal during nights.

We finally tackled these problems in the following fashion:

- We used 89C51 microcontroller owing to its simplicity and because of no requirement of a high speed microcontroller because everything was dependent on the slow sped of the mote. Even if image data could be written and compared in the memory at high speeds, it would still not be able to be transmitted over because of a very small it receiving rate of the mote. Also we had to do away with the idea of image comparison hence making the use of AL422b unnecessary. Its cost and relatively high complexity were other reasons for this choice.

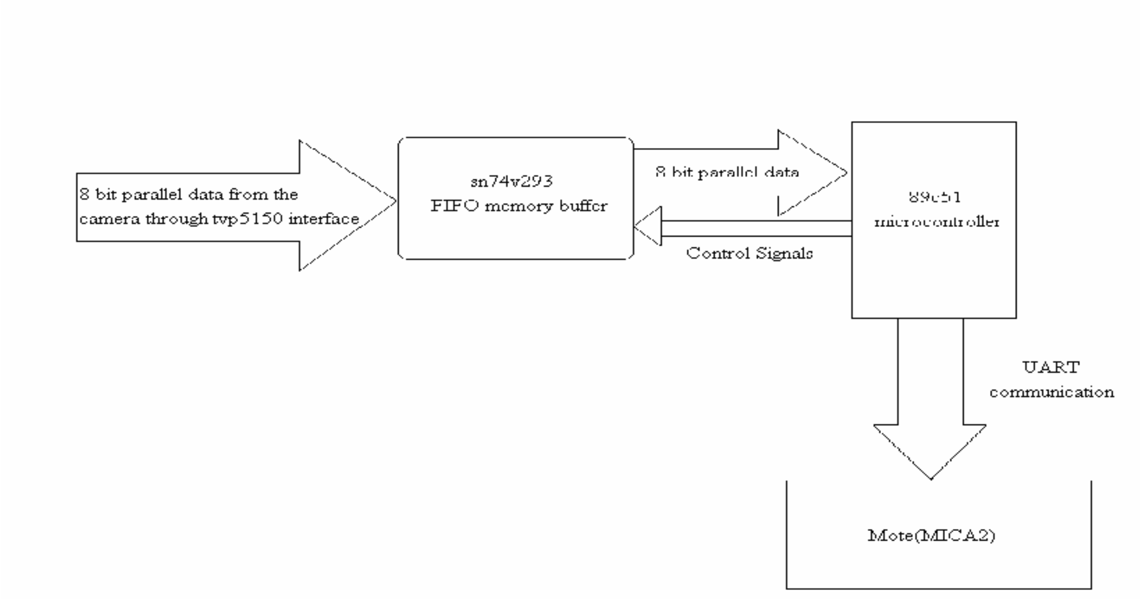
- We had to use an analog camera due to the unavailability of a suitable digital model. However, if the same if used for commercial purposes, it would greatly enhance the efficiency and will also be cost effective.

- We decided to use TI SN74V293 FIFO memory buffer. It has 128k memory with 9 bit width. Though a single chip may not be sufficient to store the full image, which is around 300k in size, Its 1st word fall through mode allows easy cascading, thereby making it suitable for our use. Furthermore its ability to operate at high speeds (it can read and write independently at up to 50 MHz, hence is easily compatible for reading the data coming from the camera.

.

- The comparison of images was not possible due to the use of FIFO memory. This meant that data is stored in a sequential fashion and we cannot jump to any particular position making it impossible to compare.

Block Diagram



Circuit Diagram

The circuit Diagram is attached.

Calculations

The baud rate of the mote is fixed at 19.2 Kilo Bauds. The serial mode 1 which is 8 bit UART with 1 start and 1 stop bit is required. The 19.2 rate is provided with the timer 1 output in mode 2. The count for the number to be loaded in the timer 2 is given by the formula:

$$N = 256 - 11.0592 \times 10^{-6} / (192 / 19200)$$

Test Procedure

After assembling the circuit, the crystals and acknowledge output are checked first to confirm the working on the µC.

Then, to check if the serial data is being transmitted, using RS232, we sent and received a file from a computer terminal to the µC via the hyper terminal. Reception of the same file that was sent confirmed the working of the code.
Then we fed in junk data in the FIFO buffer and repeated the same procedure, this time sending data from the µC.

The Problems Faced:

- The First was getting a 27 MHz synchronous clock for the memory buffer. This could not be applied through a crystal because it would be improbable to get synchronization by this approach. Hence we had to take this clock from the TVP board itself which was a difficult task because this clock was generated internally inside by the chip, whose input clock was 14.31818 MHz.

  Solution: There is a Sclk on the TVP board which provides clock at the same frequency as the pixel rate.

- Another was the problem that while packetizing the data, if 7EH byte occurred in a packet, it would seem to the MOTE that this is the end of the packet which uses 7EH as the start and stop byte.

Solution: The raw data packet uses an escape byte of 0x7D. This is needed in case a byte of payload data is the same as a reserved byte code, such as the frame synch byte 0x7E. In such a case, the payload data will be preceded by the escape byte and the payload data itself will be exclusively OR'ed with 0x20. For example, a payload data byte of 0x7E would appear in the data packet as 0x7D 0x5E.

Conclusion

A fully completed product will have the following features:

The user will get images of a remote location at regular intervals of time right on his personal computer. This is hence a very user friendly system.

This will be much cheaper than the convectional close circuit television systems and would be ideally suited for surveillance in certain cases.

Still image transmission will save more power.

Suggestions for further improvement

- Use of a digital camera instead of an analog model. This camera can be controlled to send images only when required by sending the camera in idle mode when the microcontroller is sending data to the MOTE. This will help is saving power considerably and will be help cost effective in commercial applications.

- Another power saving technique can be sending of only the error function instead of the entire image. This will also be a faster technique because now the MOTE has to transmit a considerably less amount of data. But this application will require a memory other than the FIFO used, e.g. the Averlogic 422b buffer. The error function can simply be generated by storing the 2^{nd} image and subtracting it from the previously stored 1^{st} image. This would require a buffer of at least 600 Kb.

Acknowledgement

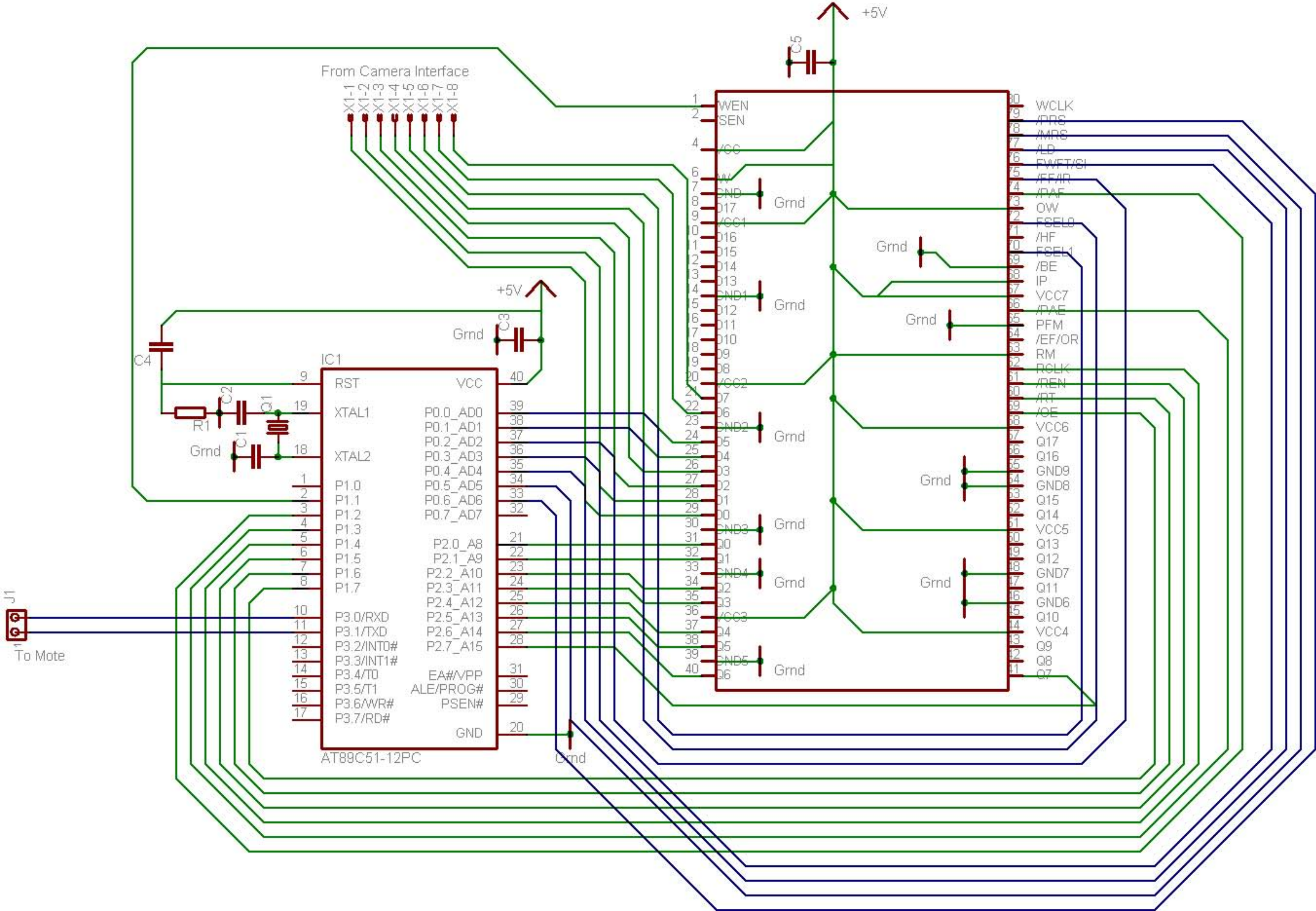
We are grateful to our project supervisor Prof. S.N. Merchant who helped us whenever the need arose. We are also especially thankful to Mr. Arvind Sharma for his generous help and kind support. We are also extremely thankful to the course instructors, mentors and the lab staff.

Users Manual

We would suggest the user to use this system only in the areas where there is not much activity observed in short intervals. Also the MOTE should be placed suitably so that it can transmit the data to the remote terminal. The system in this form operates automatically once the Vcc and Clock is given and hence there is not much sophistication involved in its use.

References

1) The data sheets of TVP 5150, 80c51, and TI SN74v293
2) Deciphering Tiny OS Serial Packets By Octave Technology
3) AT 89C51 datasheet
4) Notes on MOTE as provided by the sources from the Spann Lab.


+5V
C5
From Camera Interface
X1-1
X1-2
X1-3
X1-4
X1-5
X1-6
X1-7
X1-8
WEN
SEN
/VCC
W
GND
VCC1
GND1
VCC2
GND2
GND3
GND4
VCC3
GND5
WCLK
/PRS
/MRS
/LD
FWFT/SI
/FF/IR
/PAF
OW
FSEL0
/HF
FSEL1
/BE
IP
VCC7
/PAE
PFM
/EF/OR
RM
RCLK
/REN
/RT
/OE
VCC6
GND9
GND8
VCC5
GND7
GND6
VCC4
Grnd
C3
C4
IC1
RST
VCC
XTAL1
XTAL2
R1
C2
Q1
C1
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0_AD0
P0.1_AD1
P0.2_AD2
P0.3_AD3
P0.4_AD4
P0.5_AD5
P0.6_AD6
P0.7_AD7
P2.0_A8
P2.1_A9
P2.2_A10
P2.3_A11
P2.4_A12
P2.5_A13
P2.6_A14
P2.7_A15
P3.0/RXD
P3.1/TXD
P3.2/INT0#
P3.3/INT1#
P3.4/T0
P3.5/T1
P3.6/WR#
P3.7/RD#
EA#/VPP
ALE/PROG#
PSEN#
GND
AT89C51-12PC
J1
To Mote