UNC CHARLOTTE

# Data Logging Solution for Digital Signal Processors

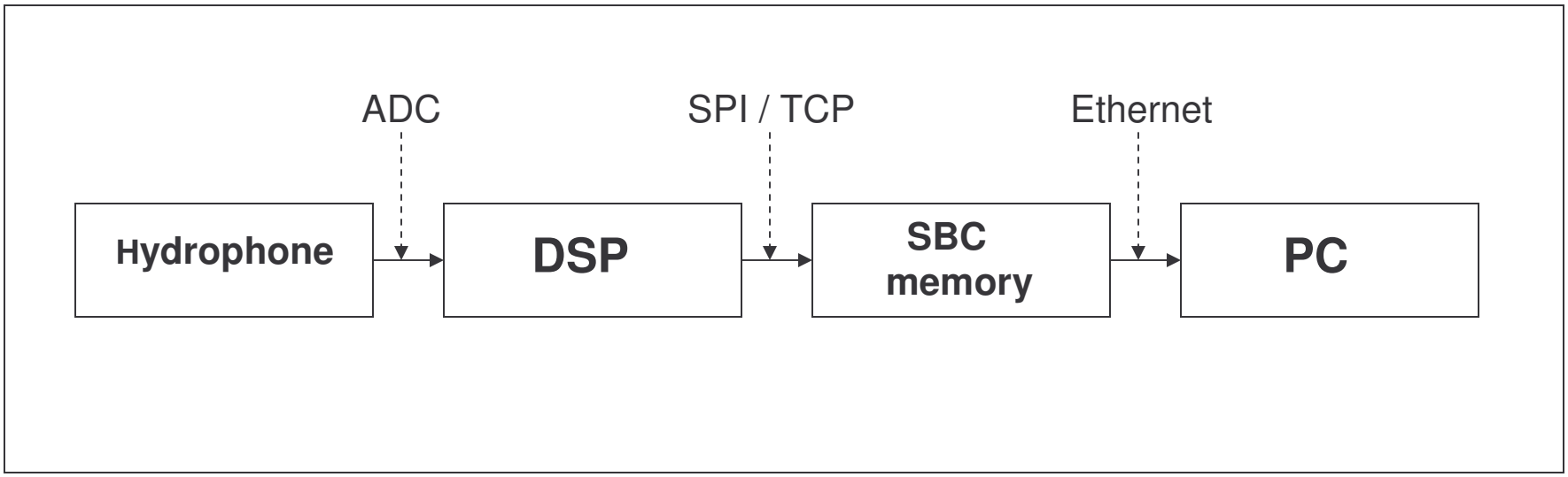
**Brian Newberry**
*Nekton Research, Inc.*
*BNewberry@NektonResearch.com*

**James M. Conrad**
*University of North Carolina at Charlotte*
*jmconrad@uncc.edu*

# Introduction

- Analyzing analog data collected by AUV (Autonomous Underwater Vehicle)
- Embedded DSP Data Logger Design for audio signal
- Digital Signal Processor (DSP) – multiple instructions per cycle
- DSP can process audio and video signals with high data rates
- DSPs consume low power

# Block Diagram

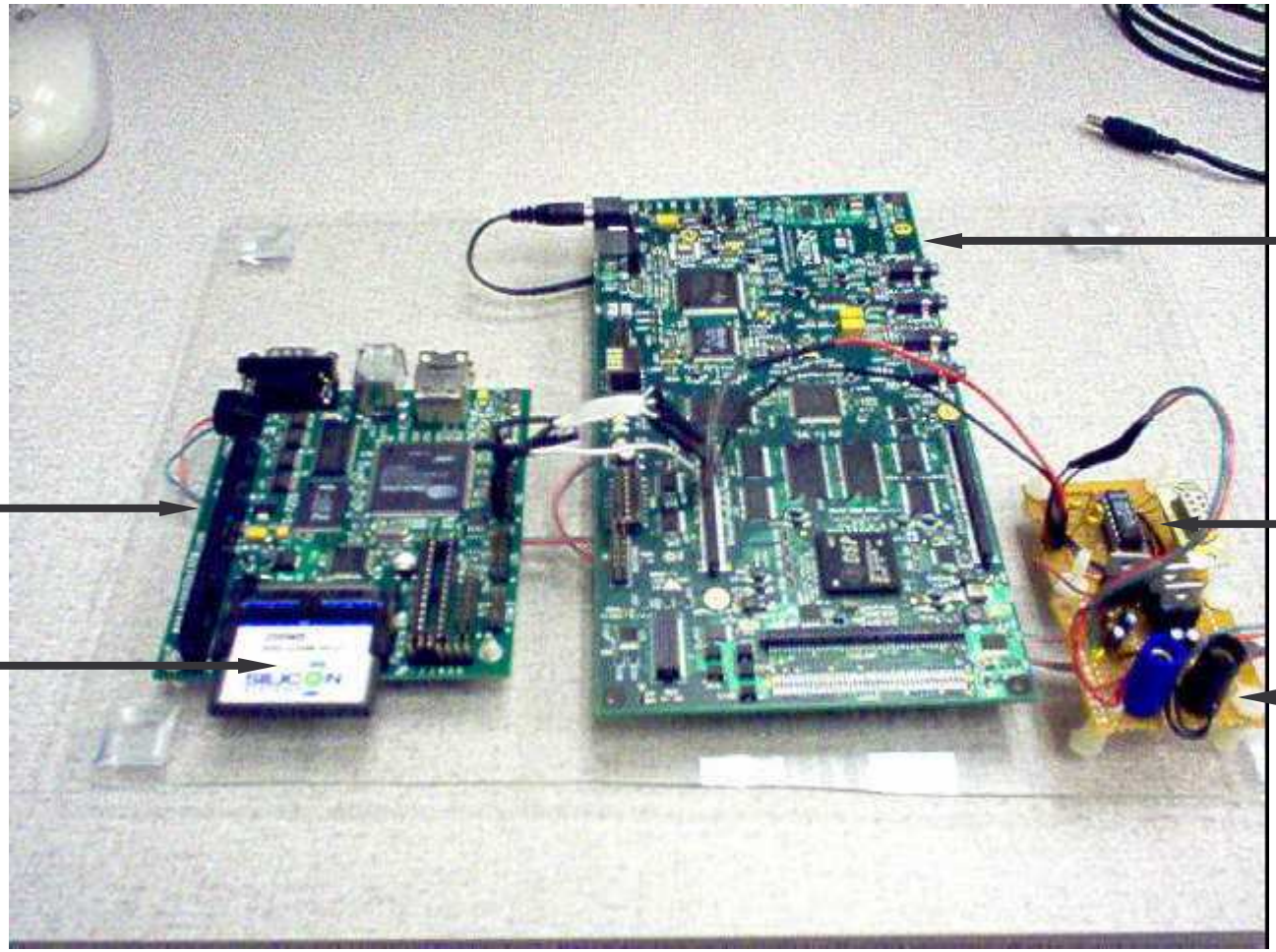


UNC CHARLOTTE

# System Design

| Component | Specification |
|---|---|
| DSP – DSK6416T | 1 MB RAM (onboard),<br>16 MB RAM (offboard),<br>512 KB Flash |
| ADC (onboard) | 96 kHz/s – sample rate (100 kHz required) |
| Compact Flash Memory Card | 256 MB, low power,<br>22 minutes data can be stored one time |
| TS-7200 Compact Flash Controller | Single board ARM based Linux machine,<br>Data rate- 1 Mbit/s, 1 mW power consumption,<br>OS- 5 MB RAM, data rate 0.85 MB/s,<br>FTP (File Transfer Protocol) server |
| RS232 (Serial Communication) | DSP communicates with hyperterminal |

# Final System Set-up



# Pseudo code (DSP operation)

Initialize system
Wait till told to start
While not told to quit
{
Gather audio data from codec
Process audio data
Report results
Send audio data to TS-7200 over SPI
Wait till reaction to reported results is complete
}

# Pseudo code (TS7200 operation)

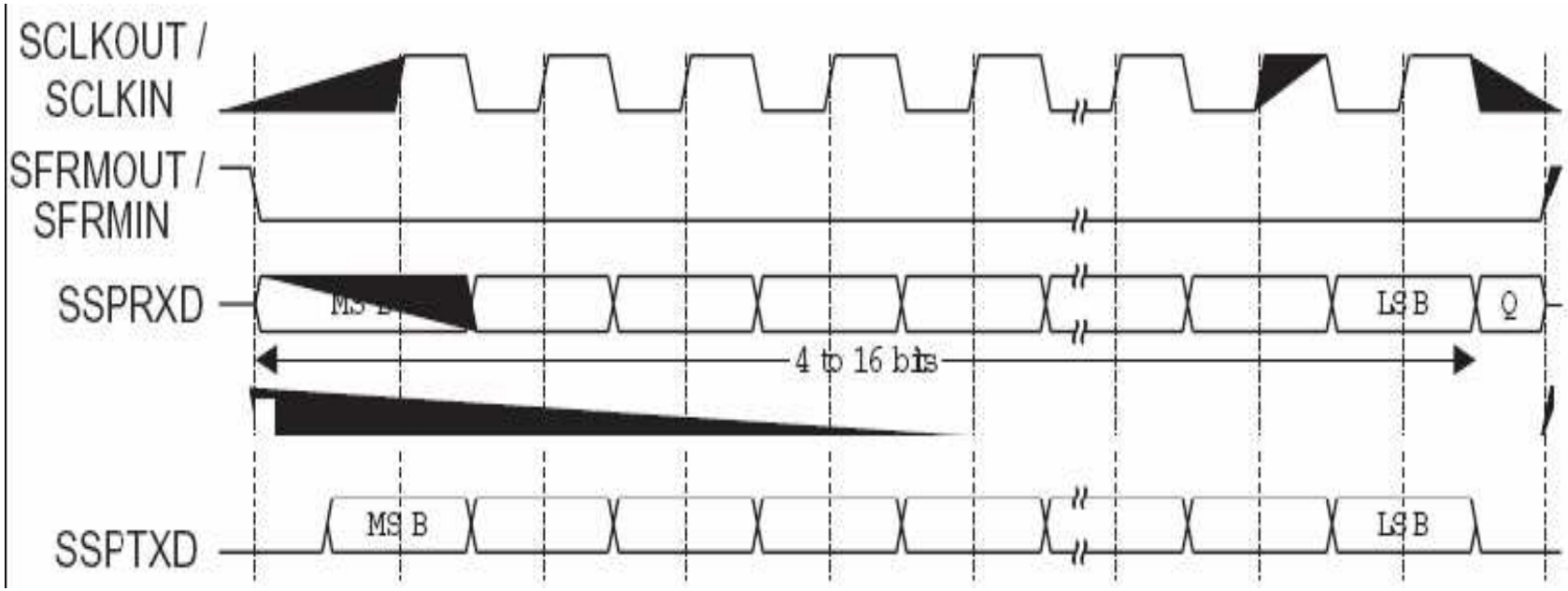
```
Initialize system
While true
{
Get Data packet
Output amount of data received
Store Data packet
}
```

UNC CHARLOTTE

# SPI Transmission Protocol

- Developed by Motorola
- Fast synchronous serial port communication
- Master-slave architecture
- DSP- master and TS7200-slave
- 1 bit from master to slave n vice versa per second
- Connections: clock, frame, ground, master-in-slave-out, master-out-slave-in

# SPI Transmission Protocol




Signal Diagram

# Characteristics of SPI Transmission Protocol

- Slave can transmit to Master only and only if Master is transmitting at the same time or else Slave has to wait
- No acknowledgement sent either by Master or Slave
- No guarantee of transmission quality
- Can change polarity of signal
- Additional bit can be induced for delay
- Zero overhead

# SPI Transmission Limitations

- Transmission speed totally dependent upon Master's clock speed
- If TS7200 would have been used as Master;
  speed range – 29 kHz to 3.7 MHz
- SPI – mainly intraboard protocol
- When used interboard – causes EMI (Electromagnetic Interference)

# TS7200 Limitations on SPI Transmission

- Linux 2.4 kernel on TS7200 - not a RTOS (Real Time Operating System)
- **Linux not a preemptive** – current task has to be finished; before starting new one
- Hence OS can not respond immediately to event occurred as DSP
- **Probability of data loss in consecutive samples**

# TCP Implementation

- To ensure, TS7200 receives data successfully from DSP
- **DSP** implementation of TCP - **Master**
  - data to slave in packets
  - special 16-bit value
  - acknowledgement from TS7200 is awaited
- **TS7200** implementation of TCP – **Slave**
  - count values received from DSP on SPI port
  - one value – correct reception
  - different value – incorrect reception

# System Performance

- Per minute - 20 seconds data is gathered, analyzed and stored
- Power consumption is minimum
  - approx. 7 watts – linear voltage regulator
  - 4.5 watts – direct power supply, no regulator
  - more efficient switching power supply needed
- If TS7200 – directly mounted over DSP board – data transfer speed would improve

# Conclusion

- DSP not burdened with memory storage – merely processes data
- Memory controller system – low power and economical
- Successful data transmission from static storage to separate PC
- DSP controls what data is logged
- System could be – stand-alone data logger
- Low cost, low power and miniature system