Name: George Yuchi
Section/Advisor: L03/Maxwell
Group Name: Team Road Rage

Microcontroller Units for Advanced Driver Assistance Systems

Introduction

With pressure mounting on automobile manufacturers to focus on vehicle safety, it's rare to not to find some form of an Advanced Driver Assistance System (ADAS) in new automobiles today. ADAS comes in various forms so as to anticipate and mitigate the various risks of driving. The different implementations of ADAS include but aren't limited to: proximity sensors, driver drowsiness detection and lane change assistance. At the heart of many ADASs is the Microcontroller Unit (MCU); the purpose of an MCU is to retrieve and process information received by the sensors of the ADAS and to ultimately deliver its interpretation to the driver. This technical review will introduce some commercially available MCUs designed for ADASs, summarize the technology of MCUs and describe a general implementation of an MCU in an ADAS.

Commercial MCUs for ADAS

The wide selection of MCUs for ADASs on the market is accounted for by the various applications MCUs are capable of handling. For example, complex workloads involved with automotive vision control call for high enough processing power to handle the Digital Signal Processing (DSP) algorithms involved with vision control [5]. The C2000 MCU family from Texas Instruments (TI) fits the automotive vision control's requirement of MCUs with high processing power. The MCUs of the C2000 family offer clock frequencies from 200 MHz to 300 MHz and comes with a special type of microprocessor architecture designed to be favorable for DSP workloads; units from this family cost from $15 to $20 [7]. The Qorivva MPC56xx family of MCUs by Freescale offers MCUs with clock frequencies from 200-264 MHz and includes FlexRay, a communication protocol that allows MCUs to communicate with each other faster than traditional protocols but at a higher cost; the MPC56xx family ranges anywhere from $10 to $20 per unit depending on the clock speed and ram size [7].

Most MCU manufacturers offer software development kits or training for the customer as a resource for familiarization with the product and its development environment. The training or software is not always free, for example a workshop for one of TI's C2000 family MCUs costs $1590 [6].

MCU Technology

Microcomputers

The main component of MCUs is the microcomputer and it is usually a circuit or VLSI (Very Large Scale Integration) core that consists of the following components: a Central Processing Unit (CPU) (aka a microprocessor), registers, memory to store dedicated programs, memory for data/stack, an interrupt handler unit and a timing unit [1,2]. An MCU is the combination of all these components as well as additional interfaces such as those required for input/output (IO) device interaction packed into one chip. To begin operation of an MCU, a program must first be placed into the memory. When the MCU is turned on, the CPU retrieves and executes the instructions of the program. Registers act as multipurpose storage at a low software level with uses such as counters and pointers important to the execution of program instructions. Timing units or clocks are for operation synchronization within the MCU and an interrupt handler alerts the MCU for events that need attention [1]. When the program finishes, data can be passed through a data bus into either memory for storage or to IO devices to be externalized.

Other Aspects of MCUs

Application and program designs for MCUs must take into consideration energy efficiency since in many applications the MCUs are battery operated. MCUs are capable of different modes of operation so that the MCU can be put in a low power consumption mode until called upon by an external event. When an external event occurs, the MCU is notified and switches to a high power consumption mode to allow for enough processing power to handle the task [1].

To protect against software failures, the MCU is guarded by a Watchdog Timer (WDT); the WDT listens to the running program and resets the MCU when communication from the program has stopped or has become irregular. There is also increasing popularity in implementations of MCUs with redundancy systems or systems such as Failure Modes and Effects Analysis (FMEA) which identifies potential failures in both hardware and software [4].

Implementation

In an implementation of an MCU for an ADAS, the MCU first receives environmental data by utilizing a CMOS sensor or other sensor attached to the vehicle to collect information. The sensors then run the data through a digital signaling system such as a Low-Voltage Differential Signaling (LVDS) system or an Analog to Digital Converter (ADC) to communicate with the MCU. The MCU receives and processes the information with the program(s) programmed in memory and then disseminates the conclusion it arrives at to either an external display unit or an external signal to communicate the results to the driver [3].

An ADAS may also consist of not one but many MCUs with different or coinciding purposes that require information to be relayed between MCUs. To communicate with each other, MCUs would utilize a protocol like FlexRay or Controller Area Network (CAN); these protocols would require compatible transceivers to be attached to or built into the MCUs [3].

[1] F.E.V Pérez, R. Pallás-Areny, “Introduction to Microcontrollers,” in *Microcontrollers: Fundamentals and Applications with PIC*, Boca Raton: CRC Press, 2009, ch. 1, sec. 1, pp. 1–9. Available: http://books.google.com/books?id=j3TJrhHnFJsC&lpg=PP1&dq=microcontrollers&pg=PA9#v=onepage&q&f=false

[2] R. Kamal, “Types Selection, and Applications of Microcontrollers,” in *Microcontrollers: Architecture, Programming, Interfacing and System Design*, Singapore: Prentice Hall, 2009, ch. 1, sec. 1, pp. 2–8.
Available: http://books.google.com/books?id=hCE4j72GO0wC&printsec=frontcover&source=gbs_atb#v=onepage&q&f=false

[3] TI, “Automotive Vision Control,” Aug. 3 2011. [Online], Available: http://www.ti.com/solution/automotive_vision_control. [Accessed Sept. 4 2011].

[4] M. Bellotti, R. Mariani, “How future automotive functional safety requirements will impact microprocessors design,” Microelectronics Reliability, vol. 50, no. 9-11, Sept. – Nov., 2010, pp. 1320-1326

[5] *Hardware Design Guidelines for TMS320F28xx and TMS320F28xxx DSC*, Texas Instruments, Dallas, TX, Aug. 2011. [Online], Available: http://www.ti.com/lit/an/spraas1b/spraas1b.pdf. [Accessed Sept. 3 2011].

[6] TI, “In-depth Design and Programming with C2000 Microcontrollers,” Aug. 6 2011. [Online], Available: http://focus.ti.com/docs/training/catalog/events/event.jhtml?sku=4DW102028. [Accessed Sept. 4 2011].

[7] Avnet, “Avnet Electronics Marketing,” Sept. 4 2011. [Online], Available: http://www.avnet.com/. [Accessed Sept. 4 2011].