Automotive audio boosters

Rein Kadastik

05.12.2019

Topics

- About me
- Car audio amplifier – whats easy, whats hard.
- Class D amplifier
- Briefly about SMPS
- Piccolo C2000 microcontroller from Texas Instruments
- Programming controllers, ADC, PWM
- Quality Assurance
- What can possibily go wrong?

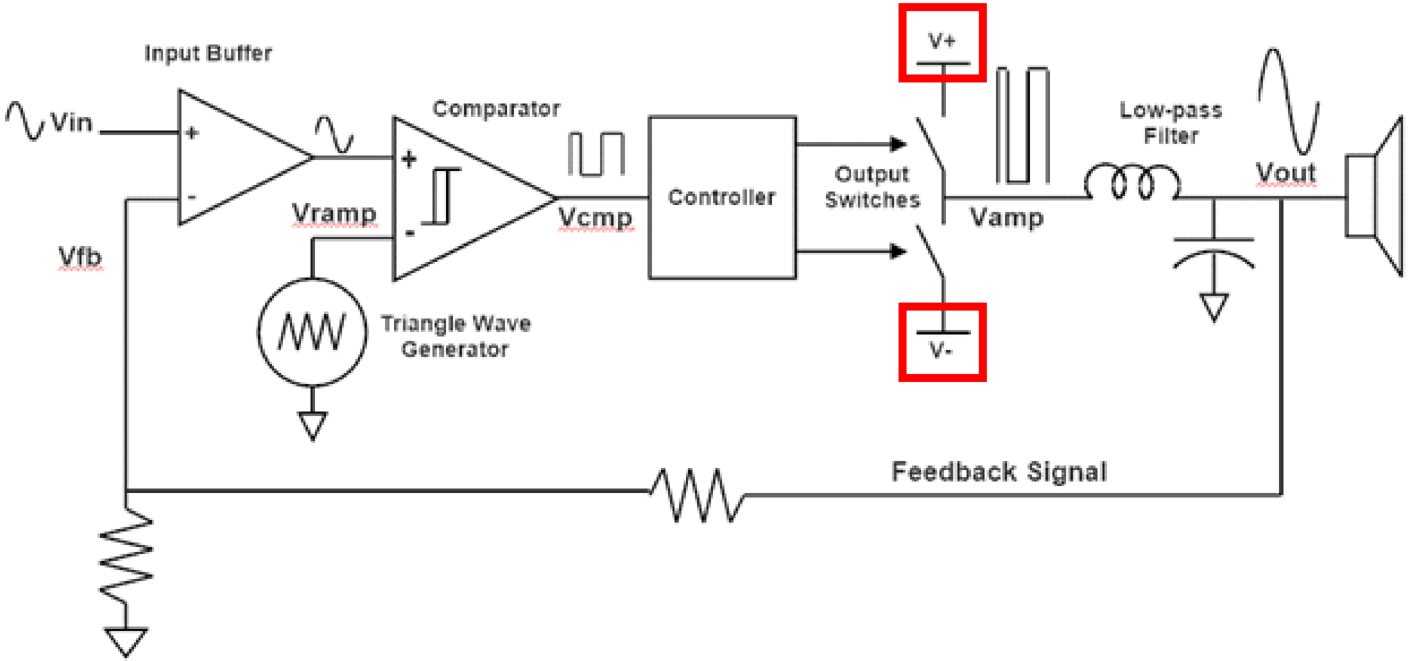
About me

- 25 years of experience in IT
- Masters degree 2007 from TTU
- Have worked on all kinds of IT projects
  - Started as sysadmin
  - Backend, databases, webservers etc
  - Frontend a little
  - Lots of integration
  - Mobile apps for iOS
  - Embedded development for last 2 years

Car audio amplifier

- Sits between the „Radio“ / headunit and the speakers.
- It’s always a class D amplifier.
- Plays audio - EASY
- Hard parts
    - Remains quiet at all other times.
    - Pass the long and rigorous testing.
    - Survive the human capability to wire everything wrong.
    - Software cannot be updated later.

Class D amplifier



https://www.whatsbestforum.com/threads/class-d-amplifiers-101.28375/

From audio to PWM

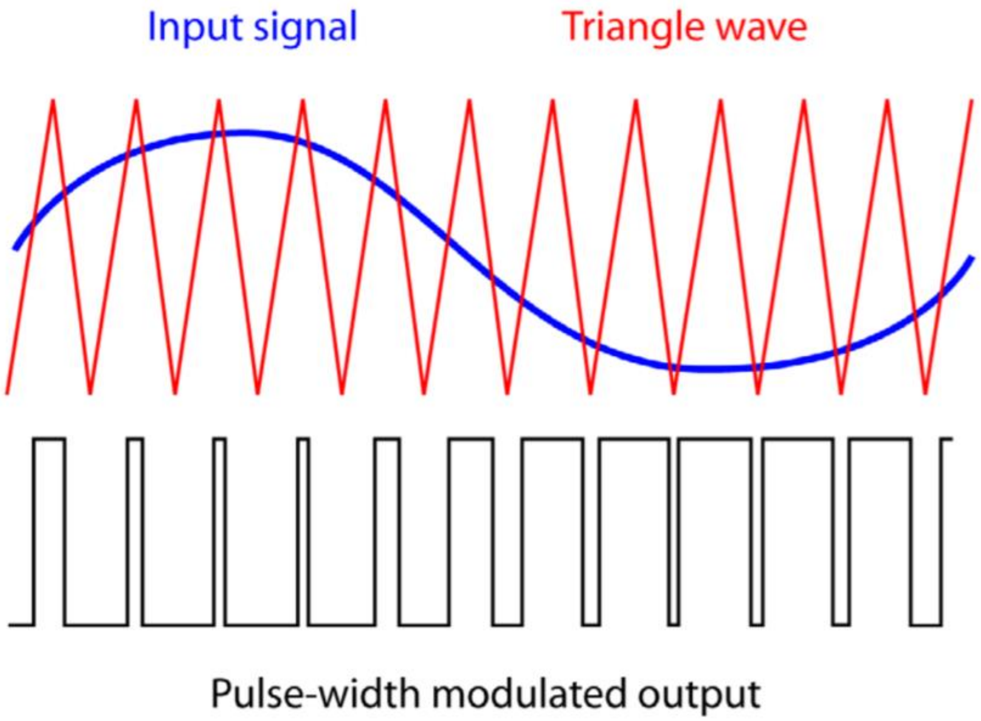
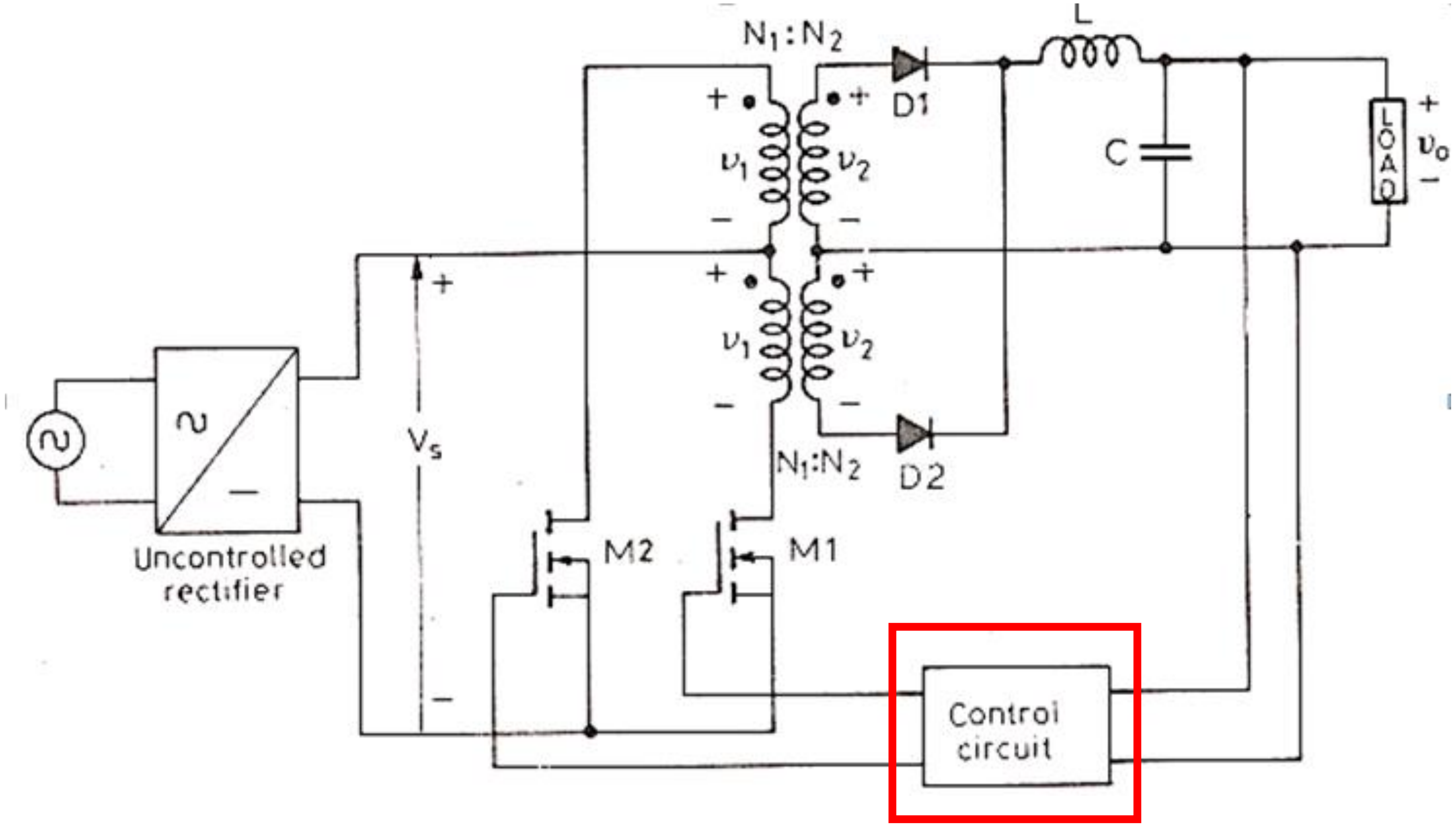


Fig. 7: Pulse-width modulation

https://www.soundonsound.com/techniques/what-class-d-amplification

SMPS push-pull topology



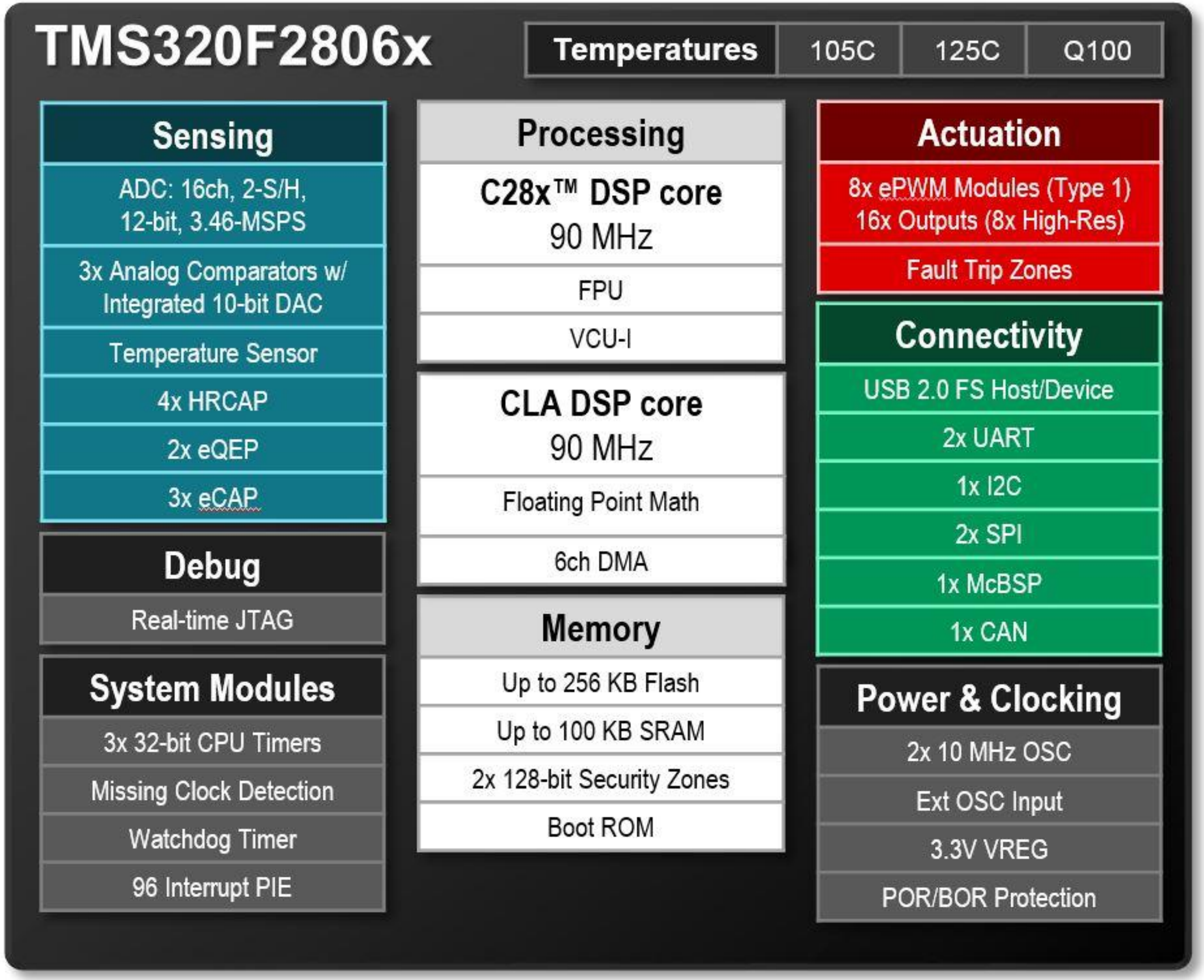
https://www.ques10.com/p/7333/use-of-power-electronic-systems-in-smps-1/

How hard can it be?

- 10% to get it working
- 90% to get it fault tolerant.

Piccolo C2000 microcontroller

purpose-built to control power electronics and provide advanced digital signal processing in industrial and automotive applications



TMS320F2806x

Temperatures | 105C | 125C | Q100

Sensing
ADC: 16ch, 2-S/H, 12-bit, 3.46-MSPS
3x Analog Comparators w/ Integrated 10-bit DAC
Temperature Sensor
4x HRCAP
2x eQEP
3x eCAP

Debug
Real-time JTAG

System Modules
3x 32-bit CPU Timers
Missing Clock Detection
Watchdog Timer
96 Interrupt PIE

Processing
C28x™ DSP core
90 MHz
FPU
VCU-I

CLA DSP core
90 MHz
Floating Point Math
6ch DMA

Memory
Up to 256 KB Flash
Up to 100 KB SRAM
2x 128-bit Security Zones
Boot ROM

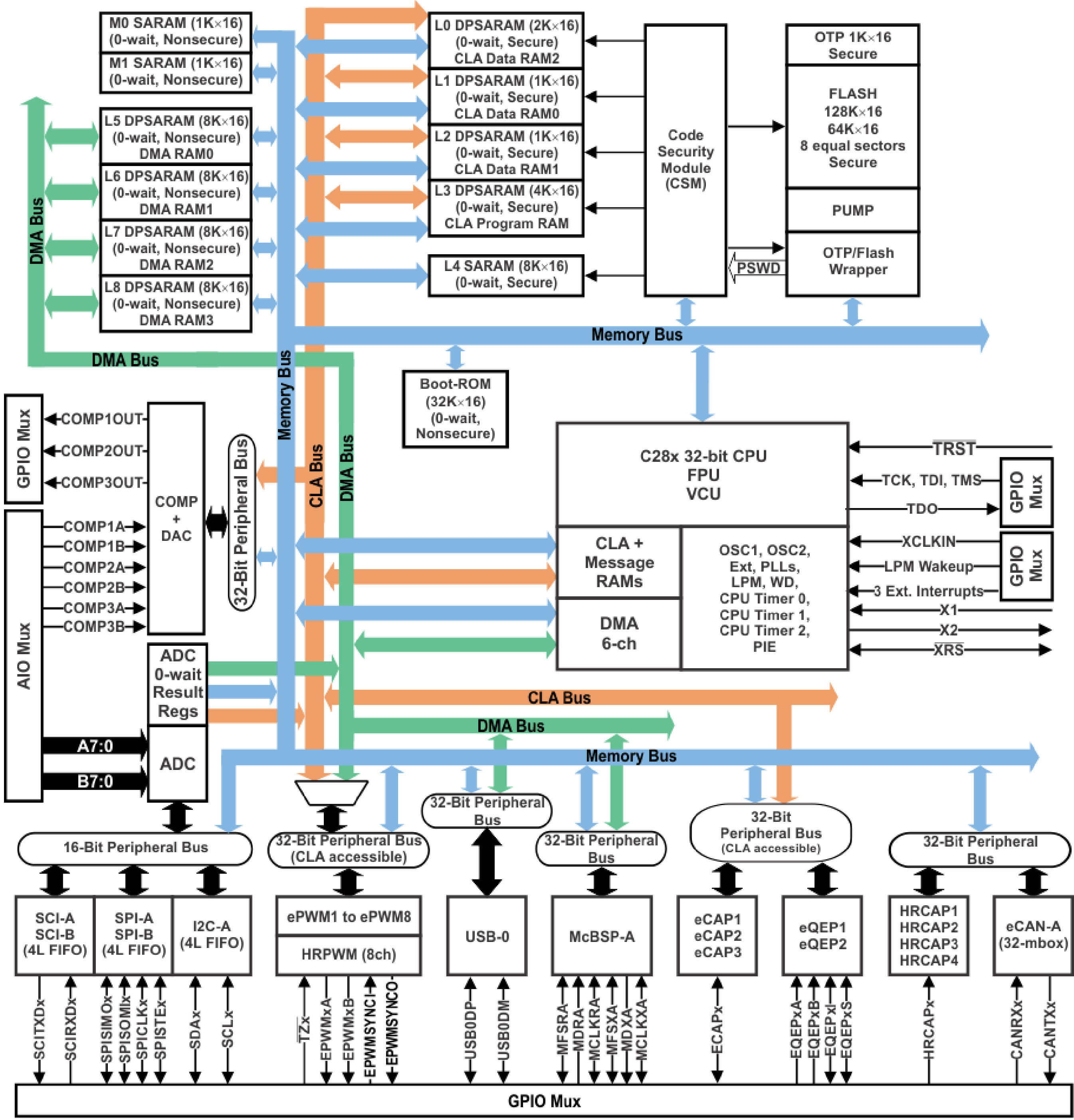
Actuation
8x ePWM Modules (Type 1)
16x Outputs (8x High-Res)
Fault Trip Zones

Connectivity
USB 2.0 FS Host/Device
2x UART
1x I2C
2x SPI
1x McBSP
1x CAN

Power & Clocking
2x 10 MHz OSC
Ext OSC Input
3.3V VREG
POR/BOR Protection

Piccolo C2000 microcontroller

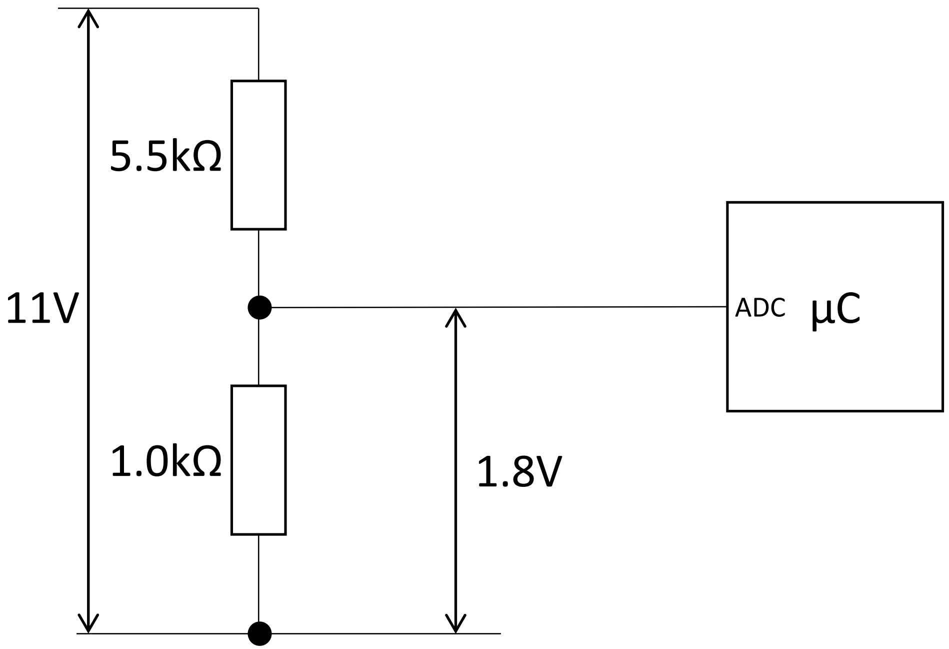
purpose-built to control power electronics and provide advanced digital signal processing in industrial and automotive applications


M0 SARAM (1K×16) (0-wait, Nonsecure)
M1 SARAM (1K×16) (0-wait, Nonsecure)
L5 DPSARAM (8K×16) (0-wait, Nonsecure) DMA RAM0
L6 DPSARAM (8K×16) (0-wait, Nonsecure) DMA RAM1
L7 DPSARAM (8K×16) (0-wait, Nonsecure) DMA RAM2
L8 DPSARAM (8K×16) (0-wait, Nonsecure) DMA RAM3
L0 DPSARAM (2K×16) (0-wait, Secure) CLA Data RAM2
L1 DPSARAM (1K×16) (0-wait, Secure) CLA Data RAM0
L2 DPSARAM (1K×16) (0-wait, Secure) CLA Data RAM1
L3 DPSARAM (4K×16) (0-wait, Secure) CLA Program RAM
L4 SARAM (8K×16) (0-wait, Secure)
Code Security Module (CSM)
OTP 1K×16 Secure
FLASH 128K×16 64K×16 8 equal sectors Secure
PUMP
OTP/Flash Wrapper
PSWD
DMA Bus
Memory Bus
CLA Bus
Boot-ROM (32K×16) (0-wait, Nonsecure)
GPIO Mux
COMP1OUT
COMP2OUT
COMP3OUT
COMP + DAC
32-Bit Peripheral Bus
AIO Mux
COMP1A
COMP1B
COMP2A
COMP2B
COMP3A
COMP3B
ADC 0-wait Result Regs
ADC
A7:0
B7:0
C28x 32-bit CPU FPU VCU
CLA + Message RAMs
DMA 6-ch
OSC1, OSC2, Ext, PLLs, LPM, WD, CPU Timer 0, CPU Timer 1, CPU Timer 2, PIE
TRST
TCK, TDI, TMS
TDO
XCLKIN
LPM Wakeup
3 Ext. Interrupts
X1
X2
XRS
16-Bit Peripheral Bus
32-Bit Peripheral Bus (CLA accessible)
32-Bit Peripheral Bus
SCI-A SCI-B (4L FIFO)
SPI-A SPI-B (4L FIFO)
I2C-A (4L FIFO)
ePWM1 to ePWM8
HRPWM (8ch)
USB-0
McBSP-A
eCAP1 eCAP2 eCAP3
eQEP1 eQEP2
HRCAP1 HRCAP2 HRCAP3 HRCAP4
eCAN-A (32-mbox)
SCITXDx
SCIRXDx
SPISIMOx
SPISOMIx
SPICLKx
SPISTEx
SDAx
SCLx
TZx
EPWMxA
EPWMxB
EPWMSYNCI
EPWMSYNCO
USB0DP
USB0DM
MFSRA
MDRA
MCLKRA
MFSXA
MDXA
MCLKXA
ECAPx
EQEPxA
EQEPxB
EQEPxI
EQEPxS
HRCAPx
CANRXx
CANTXx

Programming methods

- No libraries
- No OS
- Directly on bare metal.
- All functionality available via registers.
- Programming languages: C, assembler.
- Main task – setup and configure the microcontroller, implement interrupts.

Analog to digital converter (ADC)



Voltage divider ratio 6.1 : 1

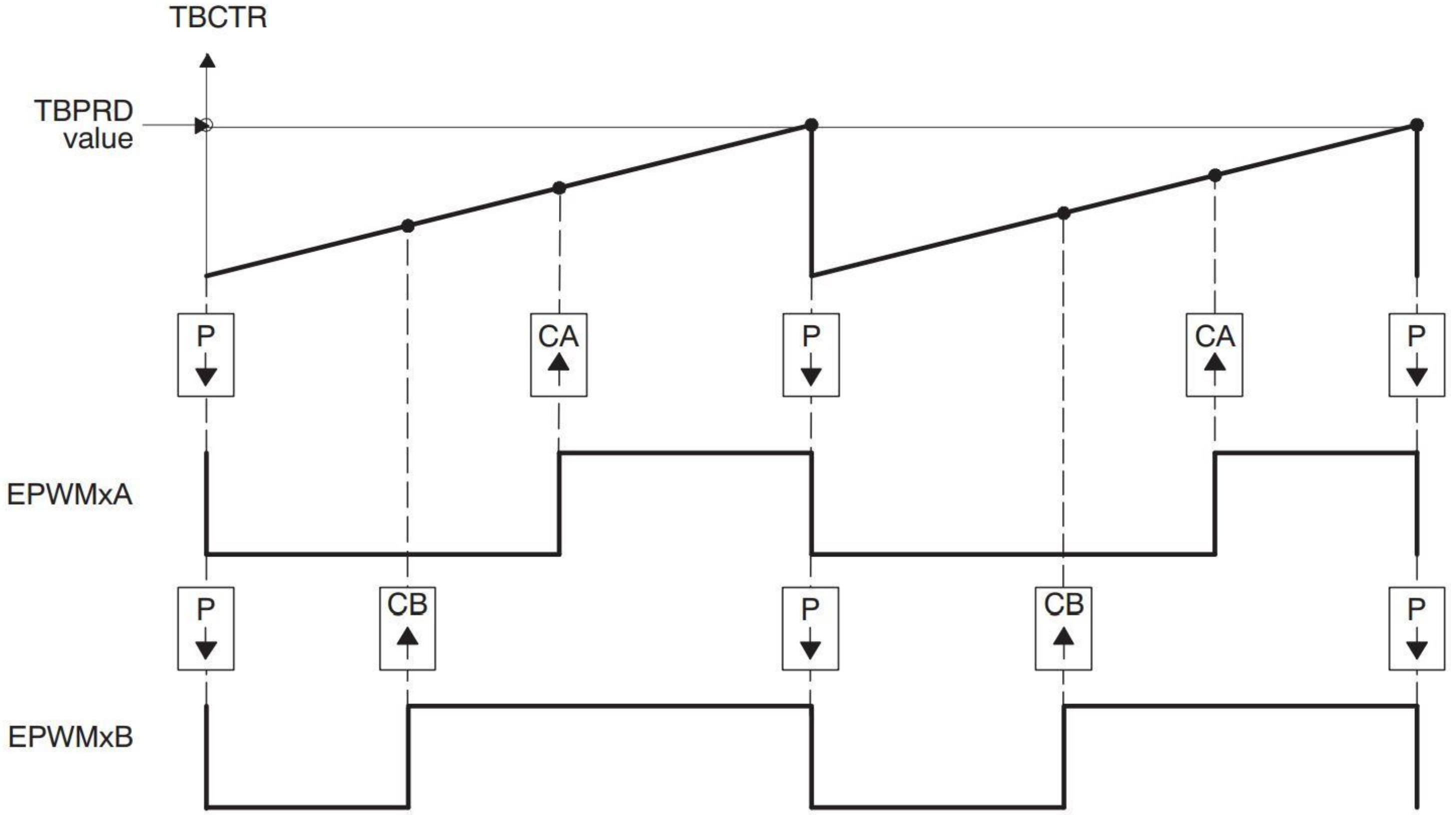
ADC voltage = 11 / 6.1 = 1.8V

ADC full range is 0 – 3.3V
12-bit ADC range mapped to 0 – 4096

1.8V / 3.3V = 54.54% of full range

ADC value: 54.54% * 4096 = 2234

Pulse Width Modulation (PWM)



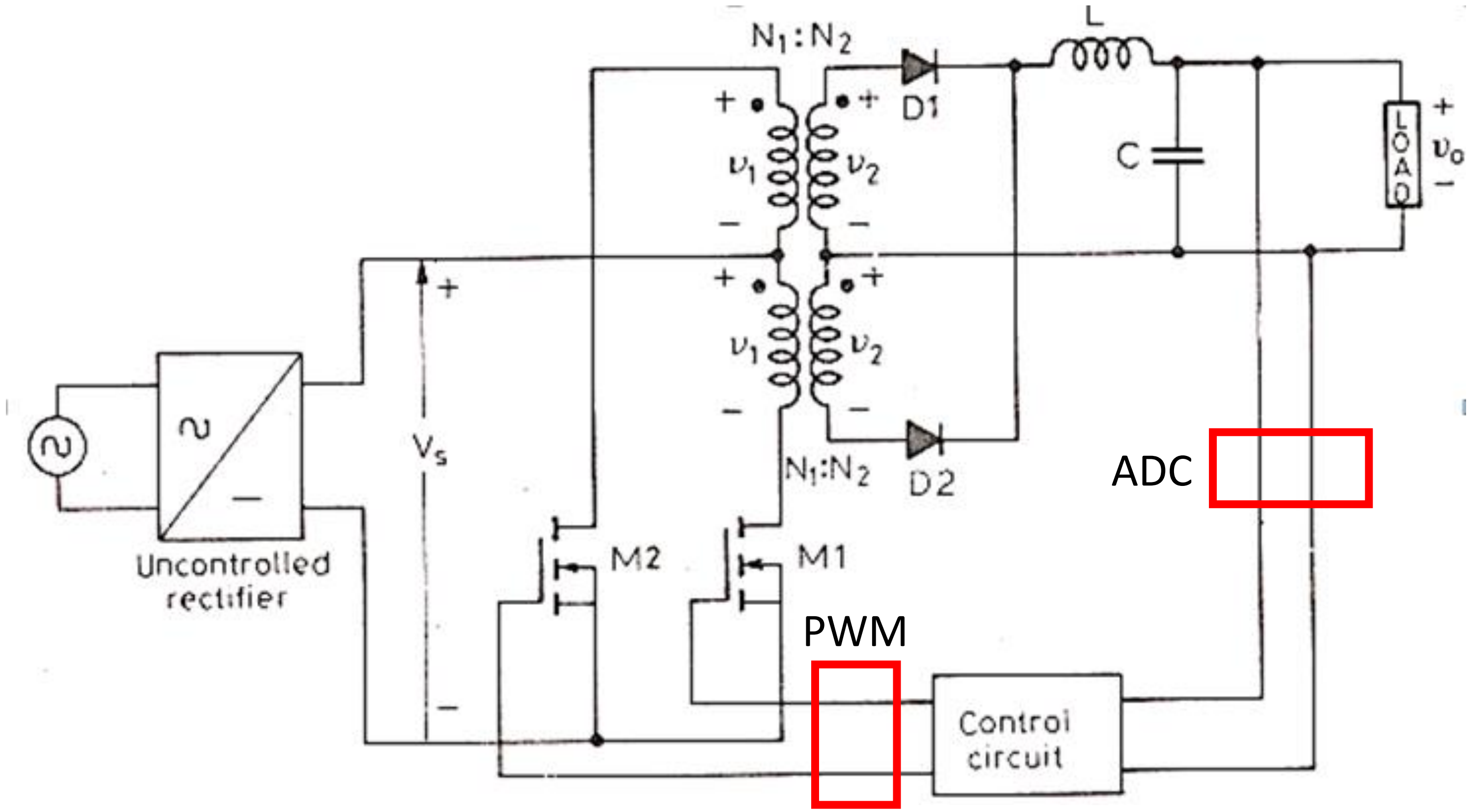
400kHz PWM @ 90MHz CPU

$$TBPRD = \frac{90\ 000\ 000}{400\ 000} = 225$$

40% PWM @ 400kHz

$$CA = 225 - (225\ * 0.4) = 135$$

SMPS push-pull topology



https://www.ques10.com/p/7333/use-of-power-electronic-systems-in-smps-1/

Getting the quality

- Code conforms to MISRA C 2012 standard.
- Static code analysis
- Unittesting with 100% coverage.
- Dedicated testers
- Validation testing

What can go wrong?

- Overclocking.
- Missing clock detection.
- Voltage rampup and hidden feedback.
- Output is stuck.
- MOSFET linear mode.
- Old software.

Thank you

Q & A