How a Lightweight RTOS can Drive CubeSat Flight Software

Andrew Kalman

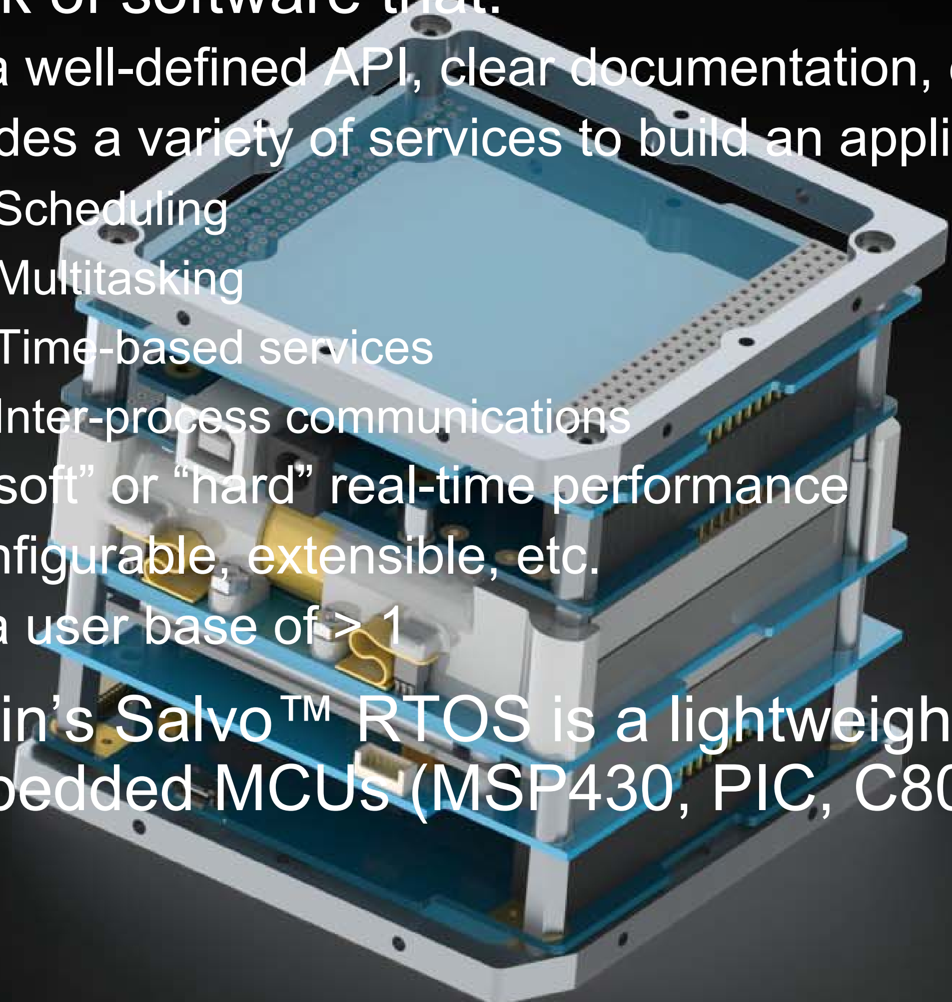
Pumpkin, Inc.

Flight Software Requirements

- Reliable - Duh
- Modular – so that multiple coders can work on it simultaneously
- (Re-)configurable – e.g. for testing, or to optimize performance.
- Efficient – use a minimum of RAM, Flash, power
- Fast – satisfy responsiveness & throughput requirements
- Capable – permit the use of (all) of the MCU's hardware / peripherals, without getting in the way
- Clean – enable a consistent programming methology

What is an RTOS?

- A chunk of software that:
  - Has a well-defined API, clear documentation, etc.
  - Provides a variety of services to build an application on top of it:
    - Scheduling
    - Multitasking
    - Time-based services
    - Inter-process communications
  - Has ‘soft” or “hard” real-time performance
  - Is configurable, extensible, etc.
  - Has a user base of > 1
- Pumpkin’s Salvo™ RTOS is a lightweight RTOS designed for embedded MCUs (MSP430, PIC, C8051, etc.)



Non-RTOS vs RTOS Coding

- Non-RTOS applications:
  - Are typically very linear in their coding and execution
  - Typically don't have scheduling, priorities, etc.
  - Their run-time performance is typically strongly affected by additions / deletions to the code
  - Must often utilize interrupts heavily to achieve a modicum of responsiveness
  - Are initially smaller … but eventually become larger than the RTOS equivalent
- RTOS applications
  - Are very loosely-coupled
  - Leverage multitasking, priorities and scheduling to maximize responsiveness, minimize load and reduce power consumption
  - Concentrate functionality via a few modules, ultimately reducing code size
  - Have a very consistent look and feel to the code

Who uses (Embedded) RTOSes?

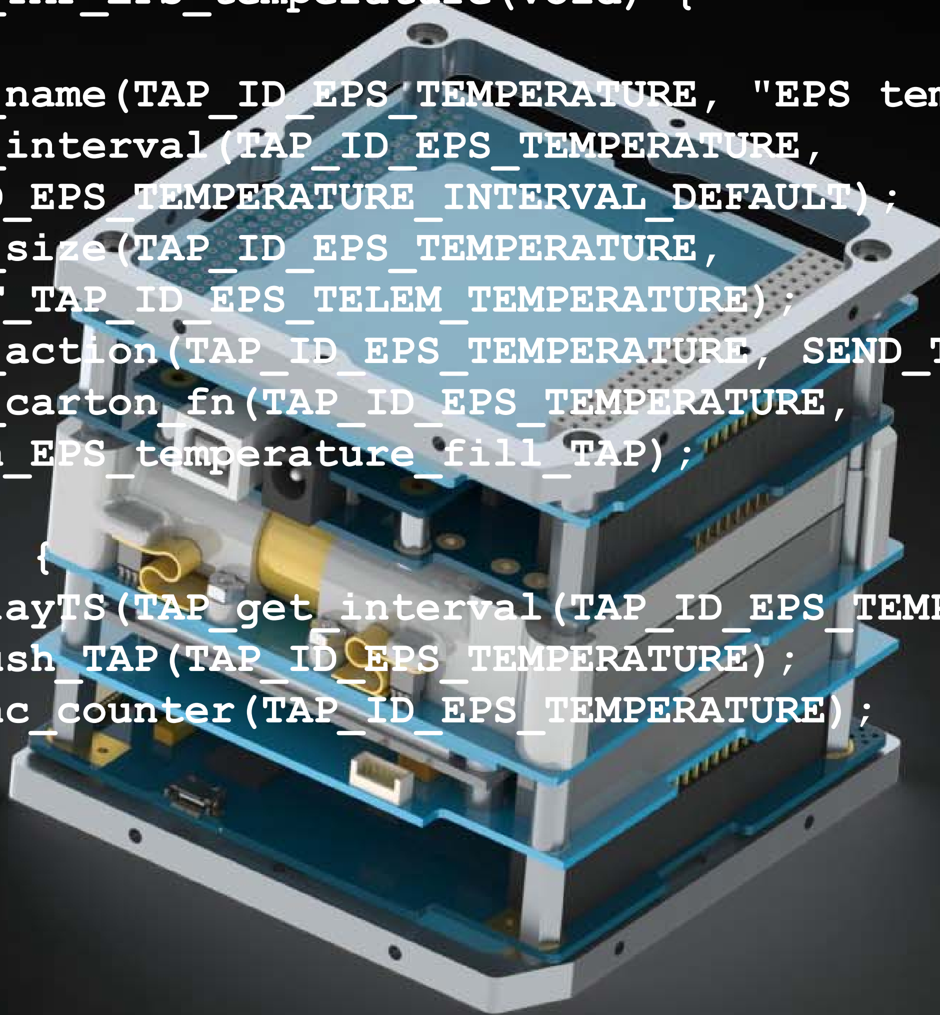
- Some Salvo RTOS applications:
  - Automated shrimp feeders in Patagonia
  - Industrial process controls
  - Health / fitness monitors
  - SDL's DICE mission
  - Sports watches
  - Bowling lanes
  - Electronic toys
  - Geotagging devices
  - Earth science sensors
  - SSDL's LMRST-Sat mission
  - All of Pumpkin's sub-Linux-size embedded controllers

Multitasking

```
OSInit();

OSCreateTask(task_cmd_do,       TASK_CMD_P,          2);
OSCreateTask(task_scpi,         TASK_SCPI_P,         1);
OSCreateTask(task_status,       TASK_STATUS_P,       3);
OSCreateTask(task_led,          TASK_LED_P,         15);
OSCreateTask(task_self_test,    TASK_SELF_TEST_P,    5);
OSCreateTask(task_vinti7,       TASK_VINTI7_P,       8);

while (1) {
  OSSched();
}
```

Periodic Task Execution

```
void task_TAP_EPS_temperature(void) {

  TAP_set_name(TAP_ID_EPS_TEMPERATURE, "EPS temperature");
  TAP_set_interval(TAP_ID_EPS_TEMPERATURE,
    TAP_ID_EPS_TEMPERATURE_INTERVAL_DEFAULT);
  TAP_set_size(TAP_ID_EPS_TEMPERATURE,
    SIZEOF_TAP_ID_EPS_TELEM_TEMPERATURE);
  TAP_set_action(TAP_ID_EPS_TEMPERATURE, SEND_TAP_SDCARD);
  TAP_set_carton_fn(TAP_ID_EPS_TEMPERATURE,
    carton_EPS_temperature_fill_TAP);

  while(1) {
    OS_DelayTS(TAP_get_interval(TAP_ID_EPS_TEMPERATURE));
    TAP_push_TAP(TAP_ID_EPS_TEMPERATURE);
    WDT_inc_counter(TAP_ID_EPS_TEMPERATURE);
  }
}
```



Waiting with a Timeout

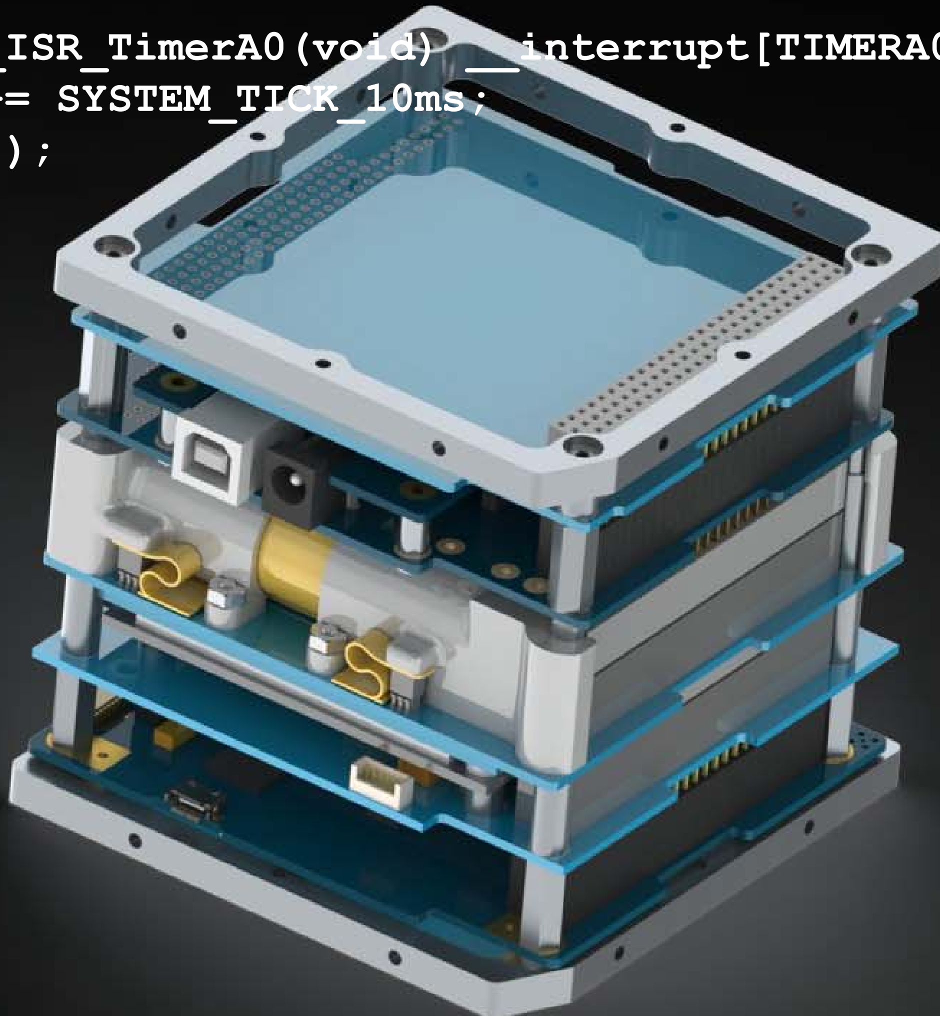
```
void task_GPS_timeout(void) {

  while(1) {
    OS_WaitBinSem(BINSEM_GPS_TIMEOUT_S_P,OSNO_TIMEOUT);
    OSTryBinSem(BINSEM_GPS_TIMEOUT_E_P);
    OS_WaitBinSem(BINSEM_GPS_TIMEOUT_E_P,GPS_TIMEOUT_TIME);
    if(OSTimedOut()) {
      gps_power(0);
    }
  }
}
```

Managing Elapsed Time

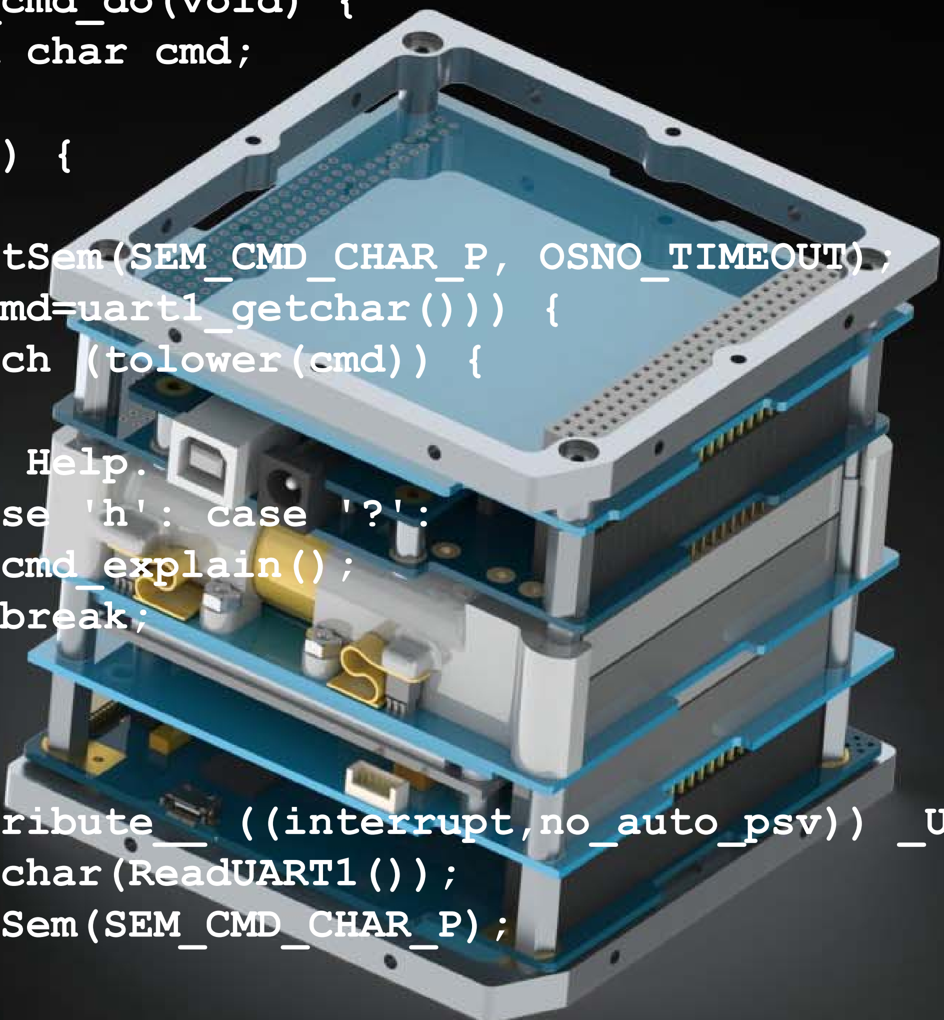
```
char * time_elapsed_DDHHMMSSTT(void) {
  OStypeTick sec, tt;
  int dd, hh, mm, ss;
  static char str[15];

  tt  = OSGetTicks();
  sec = (tt  / TICKS_PER_SEC);
  dd  = (sec /   SEC_PER_DAY);
  hh  = (sec /   SEC_PER_HOUR)
      - (dd * HOUR_PER_DAY);
  mm  = (sec /   SEC_PER_MIN)
      - (hh * MIN_PER_HOUR) - (dd *  MIN_PER_DAY);
  ss  =  sec - (mm * SEC_PER_MIN)
      - (hh * SEC_PER_HOUR) - (dd *  SEC_PER_DAY);
  tt  =  tt%TICKS_PER_SEC;
  dd %=  100;
  sprintf(str, "%02d:%02d:%02d:%02d.%02d", dd, hh, mm, ss, tt);

  return str;
}
```

All it takes is one API call …

```
void time_ISR_TimerA0(void) __interrupt[TIMERA0_VECTOR] {
  TACCR0 += SYSTEM_TICK_10ms;
  OSTimer();

  …
}
```



ISR-to-task Communications

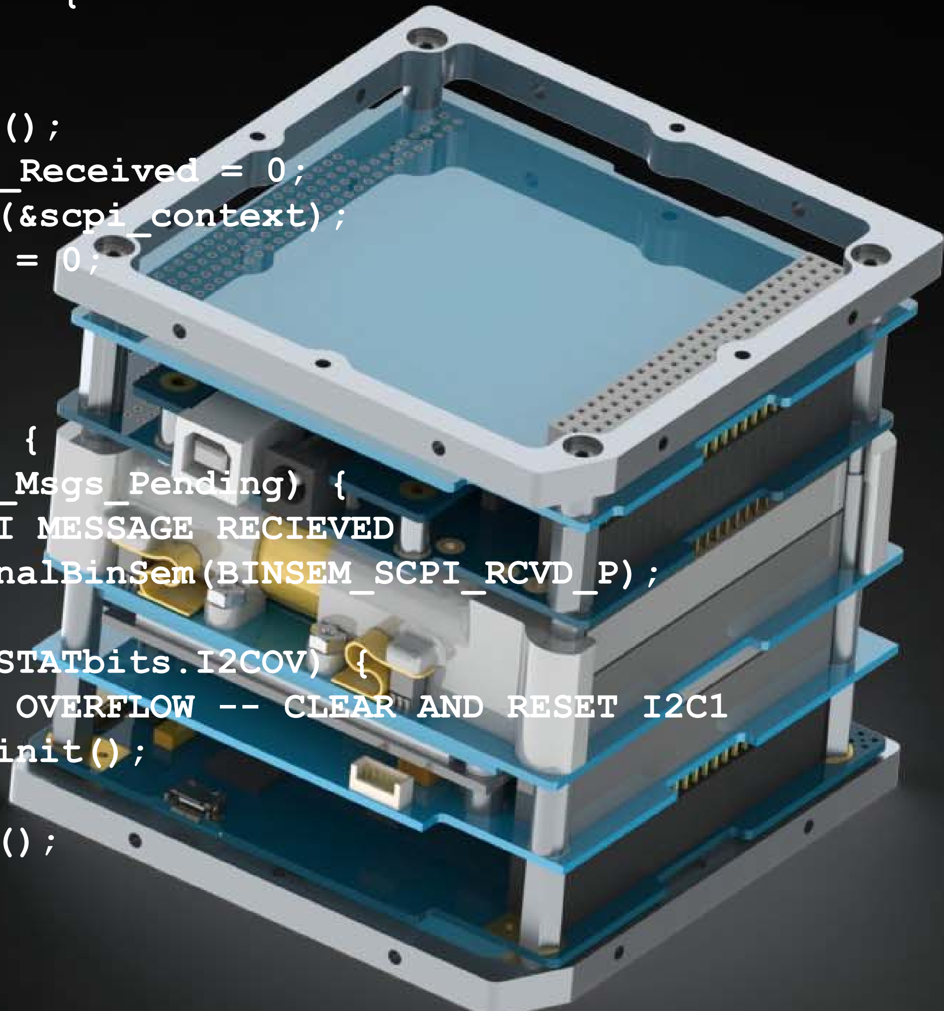


```
void task_cmd_do(void) {
  unsigned char cmd;

  while (1) {

    OS_WaitSem(SEM_CMD_CHAR_P, OSNO_TIMEOUT);
    if ((cmd=uart1_getchar())) {
      switch (tolower(cmd)) {

        // Help.
        case 'h': case '?':
          cmd_explain();
          break;
  …
}

void __attribute__ ((interrupt,no_auto_psv)) _U1RXInterrupt(void) {
  uart1_inchar(ReadUART1());
  OSSignalSem(SEM_CMD_CHAR_P);
}
```

Sleeping whenever Possible

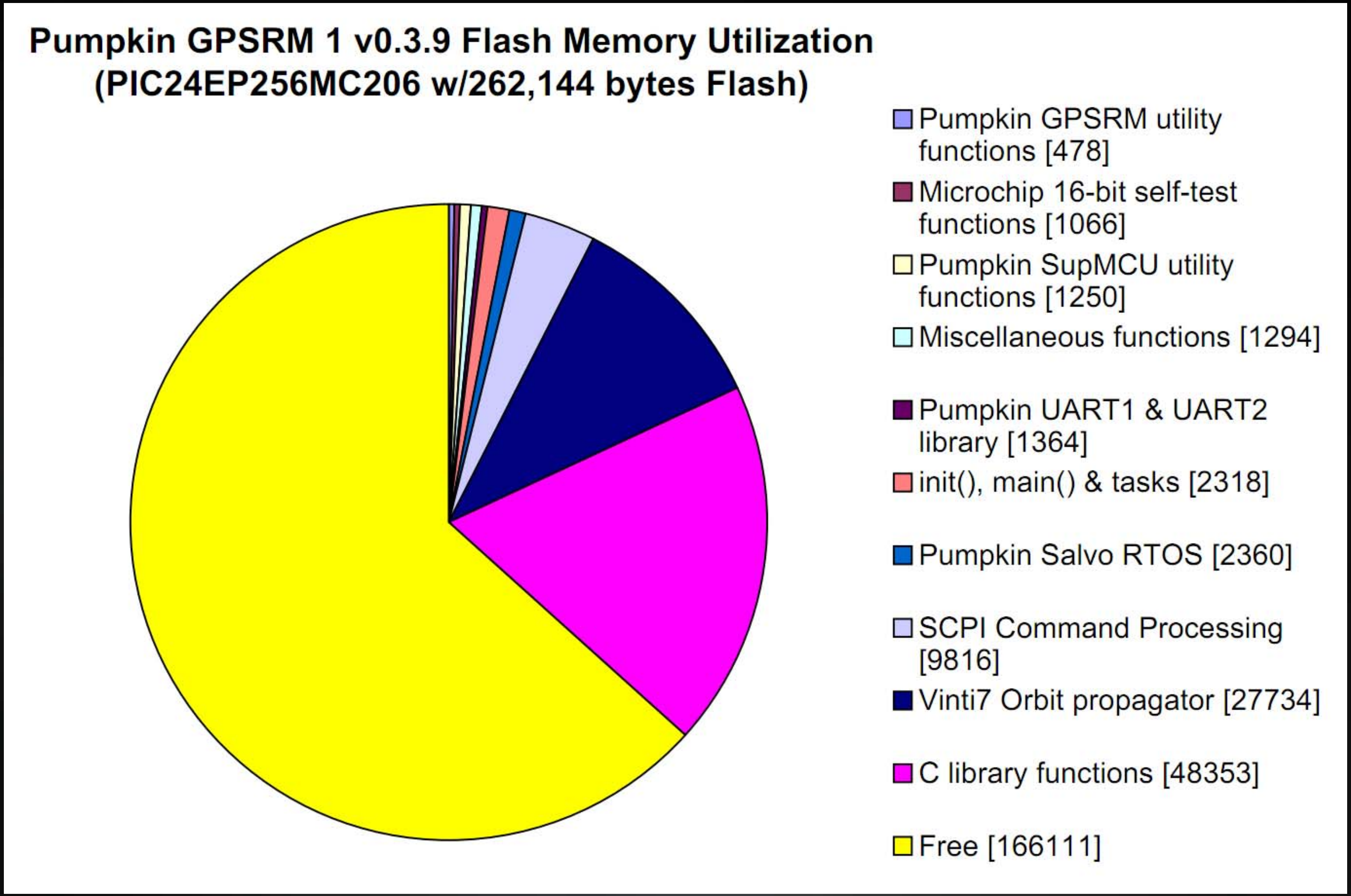
```
void OSIdlingHook(void) {
  asm(" PWRSAV #1 ");
}
```

High Runtime Performance



```
int main()    {

  init();
  i2c1_init();
  I2C1_Msgs_Received = 0;
  SCPI_Init(&scpi_context);
  scpi_cmds = 0;

  [SNIP]

  while (1) {
    if(I2C1_Msgs_Pending) {
      //SCPI MESSAGE RECIEVED
      OSSignalBinSem(BINSEM_SCPI_RCVD_P);
    }
    if(I2C1STATbits.I2COV) {
      //I2C OVERFLOW -- CLEAR AND RESET I2C1
      i2c1_init();
    }
    OSSched();
  }
}
```

Lightweight Footprint



Conclusion

- Pumpkin's lightweight Salvo RTOS has been used as the basis for flight software on multiple successful CubeSat missions
- A well-designed lightweight RTOS
  - Can have minimal impact on Flash and RAM
  - Can be exceptionally robust (see spaceflight heritage), in part because of its simplicity
  - Provides a wealth of useful features
  - Is conducive to team-based software development
  - Does not "get in the way" of real-time performance

Q&A Session

Thank you for attending this Pumpkin presentation at the 2015 CubeSat Developers' Summer Workshop!

Notice

This presentation is available online at:

www.pumpkininc.com/content/doc/press/20150808_Pumpkin_CSDWLU_2015.pdf

Appendix

- Speaker information
  - Dr. Kalman is Pumpkin's president and chief technology architect. He entered the embedded programming world in the mid-1980's. After co-founding Euphonix, Inc – the pioneering Silicon Valley high-tech pro-audio company – he founded Pumpkin, Inc. to explore the feasibility of applying high-level programming paradigms to severely memory-constrained embedded architectures. He is the creator of the Salvo RTOS and the CubeSat Kit. He holds several United States patents. He is a consulting professor in the Department of Aeronautics & Astronautics at Stanford University and directs the department's Space Systems Development Laboratory (SSDL). Contact Andrew at aek@pumpkininc.com.
- Acknowledgements
  - Pumpkin's Salvo, CubeSat Kit, MISC and SUPERNOVA customers, whose real-world experience with our products helps us continually improve and innovate.
- CubeSat Kit information
  - More information on Pumpkin's CubeSat Kit can be found at http://www.cubesatkit.com/. Patented and Patents pending.
- Copyright notice

First presented at the Pre-Conference CubeSat Workshop in Logan, Utah on Sunday, August 8, 2015, prior to the 29th Annual AIAA/USU Conference on Small Satellites.