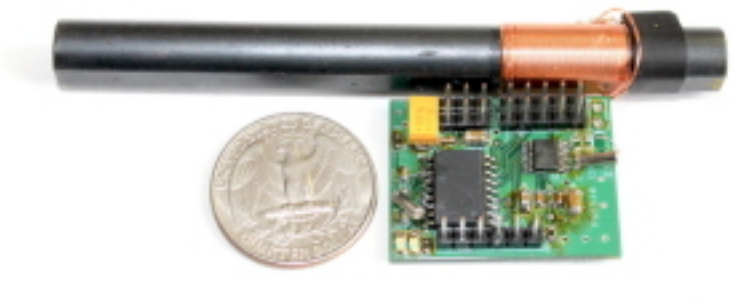


Considering its central importance to sensor networks, time synchronization has received extensive attention by the research community. Nevertheless, we note that existing approaches introduce undesirable trade-offs. For example, while GPS offers excellent accuracy for outdoor deployments, the high cost and power consumption of GPS receivers make them prohibitive to many applications. Message-passing protocols, such as FTSP, introduce different sets of compromises and constraints. In this project, we build an inexpensive and ultra-low power (Universal Time Signal Receiver
, that leverages the availability of time signals transmitted by dedicated radio stations around the globe to provide access to UTC time with millisecond-level accuracy. The proposed universal time signal receiver achieves global time synchronization and for applications where millisecond-level precision is sufficient, it consumes up to 1,000 times less energy than GPS or FTSP.

Pervasive Time Signals

Various signals have been created throughout history to serve the purpose of disseminating time information, evolving from sound and visible light to radio transmissions with increasingly sophisticated modulation techniques. Today, GPS is by far the most well known radio frequency signal that, while mainly used to perform localization, provides very accurate time information. Moreover, dozens of radiostations across the world are dedicated to broadcasting timeinformation. Table 1 presents a non-exhaustive listof such time signal stations and their operating frequencies. Most of the stations transmit in the low (LF) to high (HF) frequency radiobands using amplitude modulation (AM). This modulation scheme simplifies the reception and decoding of the time signal, thus enabling inexpensive and energy-efficient receivers. Taking the distance to the radio station into account, the time-of-flight delay can be subtracted from the decoded time signal, in order to achieve global time synchronization. In this project we focus on two LF timesignals, WWVB and DCF77, which cover most of North America and Europe,respectively.

Table 1: Time Signal Radio Stations

Station	Location	Fr
MSF	Britain	60 kHz	1966
CHU	Canada	3330, 7850, 14670 kHz	1938
BPC	China	68.5 kHz	2007
BPM	China	5, 10, 15 MHz	1981
TDF	France	162 kHz	1986
DCF77	Germany	77.5 kHz	1959
JJY	Japan	40, 60 kHz	1999
RBU	Russia	66.66 kHz	1965-74
HBG	Switzerland	75 kHz	1966
WWV	USA	2.5, 5, 10, 15, 20 MHz	1920's
WWVB	USA	60 kHz	1963

Design a Low-power Time Signal Receiver

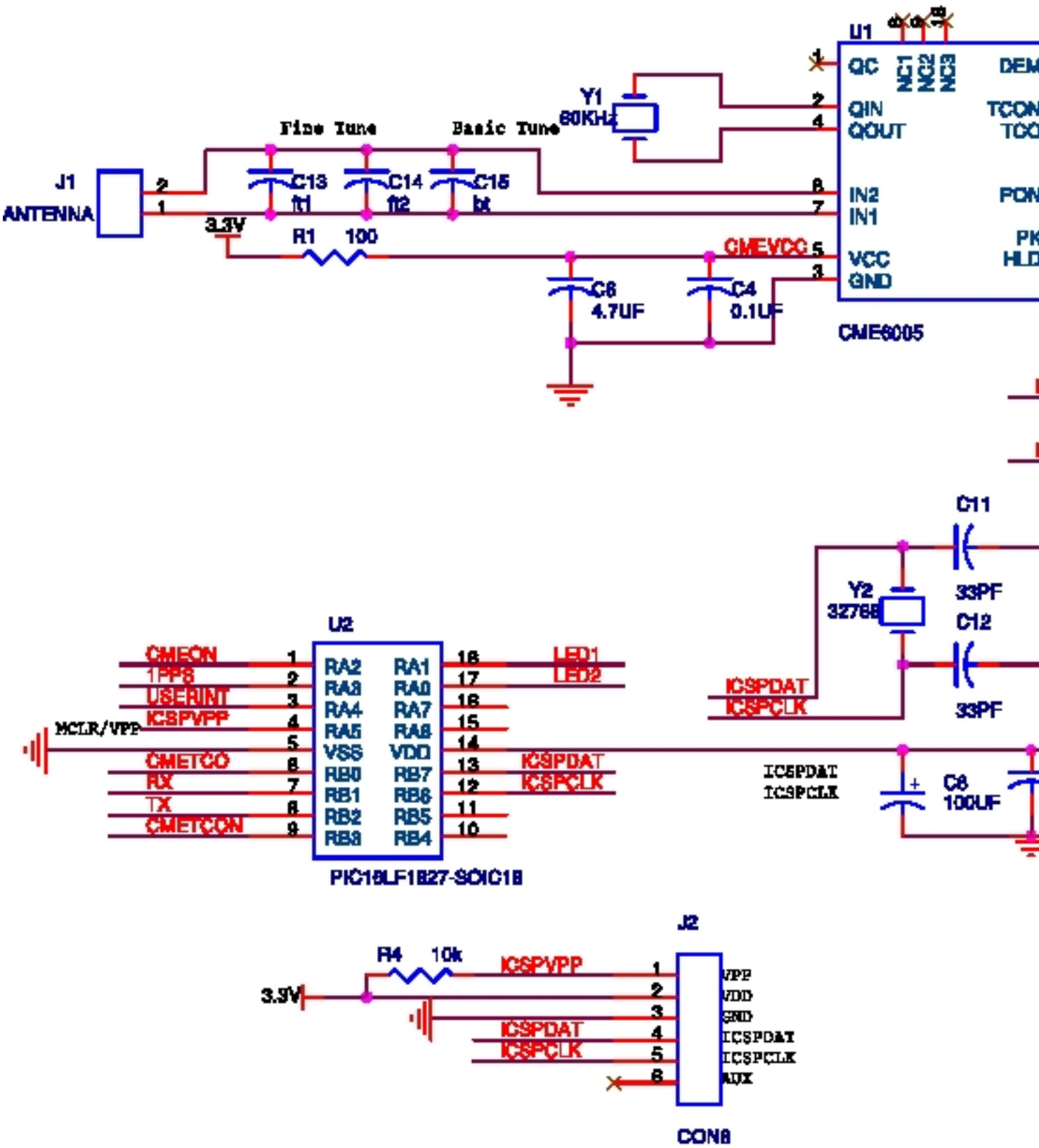
Low power consumption, high accuracy, low cost, and small form factor. These are all fundamental requirements when designing motes and by extension important design requirements for our time signal receiver as well. Specifically, low power consumption will enable liberal use of the receiver even in sensor networks with tight energy budgets, while striving for high timing accuracy is an inherent goal for all timing modules. There is however a trade-off between power consumption and timing accuracy. We decided to settle for millisecond level accuracy by leveraging the pervasive time signals, because these AM-modulated time signals can be received with inexpensive and low-power radios. Furthermore, we observe that millisecond level accuracy is actually sufficient for many sensor network applications, especially in the area of environmental monitoring, since these deployments typically measure slow changing physical values. Finally, low cost and small form factor would increase the feasibility of widely deploying this universal time signal receiver.

- **Radio chip**. In order to achieve low power consumption and small footprint, we based our prototype design on the **CME6005** radio chip from C-MAX. CME6005 has a frequency range spanning from 40 to 120 kHz and can receive the WWVB, DCF77, JJY, MSF and HBG time signals. Moreover, CME6005 offers low power conumption: less than 90 uA in active mode and 0.03 uA in standby mode.

- **MCU**. We choose the extreme low-power microcontroller **PIC16LF1827** from Microchip for decoding time signal bits output by the CME6005. This PIC consumes 600 nA in sleep mode

with a 32 kHz timer active, and 800 uA when running at 4 MHz. By connecting the CME6005's output pin to an interrupt-enabled input pin on the PIC, we allow the reception and decoding of time frames to be interrupt-based.

The schematic is shown as follows. The PDF version is available for downloading **here** .



Tmote Sky 32 kHz Clock Frequency Variation Dataset

We did two experiments to collect frequency variation data in indoor and outdoor environments. Both experiments lasted more than 7 days. This dataset is availabe **here** .

This dataset can tell us how the mote clocks might vary in real world scenarios over the course of one week. Nevertheless, we note that the outdoor experiment was in the spring and was placed under a tree in a residential park, which means that all the motes were experiencing similar ambient temperatures. We expect that in a deployment where the motes are placed at more diverse locations, clock frequency variations are likely to be much more dynamic. Therefore, this dataset provides a means to evaluate the performance of timesynchronization protocols through simulation.

Experiment Configurations

Multiple Tmote Sky motes are wired together (VCC, GND, GPIO pins) and placed in a plastic container. The mote radios (CC2420) are turned off. In the indoor experiment, 9 motes are used; in the outdoor experiment, 10 motes are used.

Both experiments are in the spring of 2010, and for the outdoor experiment, the plastic container is placed in a small park and under a tree. For the indoor experiment, the container is placed in an apartment building.

One mote is chosen as the controller, and it toggles its output GPIO pin every 3137 ticks of the TimerMilliC (1 tick = 1/1024 second, therefore 1 second = 1024 ticks).

Each of the other motes will fire an interrupt at the falling edge of the GPIO pin, therefore the interrupt will fire once for every 3137*2 ticks of the TimerMilliC on the controller mote. When the interrupt fires, each mote will read its local 32 kHz timer and write the value to the flash storage.

If the mote clock and the controller's clock have the exact same frequency (no clock skew), the 32 kHz timer reading would increment 32*3137*2=200768 ticks between two interrupts. Note that one tick of the 32 kHz timer is equal to 1/32768=30.517578125 microseconds.

Data Format

The txt file records the local 32 kHz timer readings on all the motes (excluding the controller)

when the interrupt fires. Each column corresponds to one mote. All the clocks are reset to zero when the first interrupt fires, and are then left free running for more than 7 days.

If the clocks on the motes are perfectly in sync, then the readings on the same row should be equal. Likewise, within each column, the reading should increment 32*3137*2=200768 for each row. However, the clock frequencies fluctuate in the range of tens of ppm (part-per-million) and as a result the readings on different motes will drift over time. This can easily be seen from the two txt files.

Since the interrupt is driven by the GPIO output of the controller, therefore the interval between two readings is 3137*2/1.024=6127 milliseconds, subject to the local clock skew on the controller.

Reference

- Yin Chen, Qiang Wang, Marcus Chang, Andreas Terzis, **Ultra-Low Power Time Synchronization Using Passive Radio Receivers**. To appear in the proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN/SPOTS), 2011.
PDF