Motivation
Ad-hoc sensor networks are flexible and easy to deploy...
...but their ad-hoc, reactive nature makes them difficult to program.

TinyOS is an event-driven operating system for networked sensors. Its flat hierarchy makes it difficult to predict the flow of control between synchronous and asynchronous parts of the application.

TinyOS extends nesC (the language used to specify TinyOS programs) to provide a GALS programming model that separates the rates of control of the system — the reactive part and the computational part — via asynchrony. galsC programs have minimal overhead, are easy to write and allow reuse of software components.

Software Synthesis
After the programmer specifies the actors comprising the application and the connections between actor ports, the galsC compiler generates code for the scheduling of and communication between actors.

The galsC compiler also generates access functions for TinyGUYS, access calls, and buffers for the data.

Example: Peak Finding

Future Work
- Generate from Ptolemy domain (compare to CI).
- Blocking write: retry when queue is full.
- Priority scheduling algorithm with queue insertions.
- Determine queue sizes automatically.
- Run-time reconﬁgurability of actors.